

ALMA MATER STUDIORUM · UNIVERSITÀ DI
BOLOGNA

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
Corso di Laurea Magistrale in Informatica

**Competenze e competizioni
di informatica:
valutazioni sperimentali**

Tesi di Laurea in Storia e Didattica dell'Informatica

Relatore:
Chiar.mo Prof.
Giorgio Casadei

Presentata da:
Luca Leoni

Sessione III
Anno Accademico 2011/2012

Indice

1	Introduzione	3
1.1	Motivazioni	3
1.2	Struttura della tesi	6
2	Computational thinking	7
2.1	Definizione corrente	7
2.2	Computational thinking nella didattica	15
2.3	Computational thinking e altre discipline	16
2.4	Il computational thinking nelle scuole	18
3	Progetto problem solving	23
3.1	Introduzione	23
3.2	Specifiche	24
3.3	Syllabus	25
3.4	Assi culturali	26
3.5	Regolamento	33
3.6	Le prove	34
4	Analisi di dati sperimentali	47
4.1	Valutazioni per categorie di problemi	48
4.2	Le eccellenze	58
4.3	Valutazioni generali dei risultati	59
4.4	Problemi dell'impianto e proposte	64
5	Conclusioni	69
5.1	Conclusioni	69

5.2	Sviluppi futuri	71
	Bibliografia	73

Elenco delle figure

1.1	Metafora dell'iceberg	5
3.1	Torre di Hanoi	36
3.2	Tabella delle attività	38
3.3	Esercizio di matematica: sorgenti e rigagnoli	41
3.4	Piano della tartaruga	43
4.1	Situazione generale della scuola primaria	50
4.2	<i>Italiano e inglese</i> nelle scuole primarie	50
4.3	Aggregazione di <i>italiano e inglese</i> nelle scuole primarie	51
4.4	<i>Matematica</i> nelle scuole primarie	51
4.5	<i>Grafi e knapsack</i> nelle scuole primarie	52
4.6	L'area scientifica nelle scuole primarie	52
4.7	Situazione generale della scuola secondaria di primo grado	53
4.8	<i>Italiano</i> nella scuola secondaria di primo grado	53
4.9	Aggregazione di <i>Italiano e inglese</i> nella scuola secondaria di primo grado	54
4.10	<i>Project management</i> nella scuola secondaria di primo grado	54
4.11	<i>Grafi e knapsack</i> nella scuola secondaria di primo grado	55
4.12	L'area scientifica nella scuola secondaria di primo grado	55
4.13	Situazione generale della scuola secondaria di secondo grado	56
4.14	<i>Italiano e inglese</i> nelle scuole secondarie di secondo grado	57
4.15	Categorie dell'area scientifica nelle scuole secondarie di secondo grado	57

4.16 L'area scientifica e l'area linguistica nelle scuole secondarie di secondo grado	58
4.17 Scuola primaria, studenti finalisti	60
4.18 Scuola primaria: confronto fra "bravi" e il resto degli studenti . .	60
4.19 Scuola secondaria di primo grado: studenti finalisti	61
4.20 Scuola secondaria di primo grado: confronto fra "bravi" e il resto degli studenti	61
4.21 Scuola secondaria di secondo grado: studenti finalisti	62
4.22 Scuola secondaria di secondo grado: confronto fra "bravi" e il resto degli studenti	62
5.1 Linguaggio visuale Scratch: esempio di ricerca binaria	72

Elenco delle tabelle

1.1	Contributi culturali delle discipline	5
3.1	Syllabus scuola primaria	27
3.2	Syllabus per le tre classi del ciclo di istruzione secondario di primo grado	28
3.3	Syllabus per le classi I e II del ciclo di istruzione secondario di secondo grado	29
3.4	Tabella raffigurante il numero minimo di mosse necessarie per la risoluzione di una Torre di Hanoi con n dischi.	37

Capitolo 1

Introduzione

Questa tesi prende atto che la scuola italiana non ha dell'informatica una percezione coerente con il suo statuto scientifico e culturale, così come per esempio sta emergendo nel contesto che si viene consolidando nell'ambito del cosiddetto computational thinking [1]. La ricerca svolta prende in considerazione le motivazioni, i contenuti e i risultati di un progetto del MIUR che ha coinvolto da tre anni centinaia di scuole di tutto il ciclo dell'obbligo con l'obiettivo di individuare i cambiamenti indotti dallo svolgimento di allenamenti di problem solving per una corretta percezione dei concetti del computational thinking. L'ipotesi che sostiene questo lavoro è che appropriati allenamenti di soluzioni di problemi contribuiscano non solo a migliorare competenze di problem solving, ma anche a migliorare la percezione dei concetti di informatica nel contesto del computational thinking.

1.1 Motivazioni

La scuola italiana rischia come il Titanic: naviga verso il futuro senza una chiara percezione dell'iceberg rappresentato dall'Informatica. Le scialuppe sono rappresentate da alcune lodevoli iniziative, proposte dal Ministero, cui aderiscono volontariamente le scuole grazie alla lungimiranza dei docenti e all'entusiasmo degli studenti. In particolare, in questa tesi, viene preso in considerazione il progetto del MIUR Olimpiadi di Problem Solving e vengono analizzati i risultati delle prestazioni degli studenti nelle prove di allenamento e di selezione competi-

tiva svolte negli anni scolastici 2009/2010 e 2010/2011. Questa sperimentazione ha coinvolto gli studenti della scuola dell'obbligo di oltre 650 istituti scolastici. Le ragioni che hanno spinto il MIUR ad aderire alla proposta di lanciare il progetto Problem Solving risiedono nella constatazione che l'Informatica non è percepita correttamente - ovvero come una disciplina scientifica - come lo sono, per esempio, la matematica, la fisica, o la filosofia. In generale, quindi, nella scuola sono ignorati i contributi scientifici e culturali che l'informatica può offrire grazie ai profondi legami storici con le discipline che ne costituiscono il tradizionale contenuto. La metafora dell'iceberg si presta bene per sottolineare questo aspetto. L'Informatica infatti può essere presentata come un iceberg (Figura 1.1): la parte emersa che tutti vedono rappresenta le applicazioni e i servizi oggi diffusi in ogni area applicativa indicati genericamente come TIC (Tecnologie dell'Informazione e della Comunicazione); la parte sommersa rappresenta il contenuto scientifico e culturale che consente la realizzazione di quei servizi e l'applicazione delle tecnologie. Questa metafora vale anche per altre discipline, e la scuola lo riconosce dando opportuno spazio sia alla parte emersa (i contenuti operativi) che alla parte sommersa (i contenuti scientifici e culturali), si veda Tabella 1.1. Nelle scuole sono presenti i computer, non l'Informatica. Per usare un computer sono sufficienti abilità operative, mentre per usare l'Informatica sono necessarie competenze concettuali. Non è sufficiente saper usare il computer. La scelta del progetto Problem Solving per ovviare a questa lacuna sta nella consapevolezza che l'Informatica va introdotta come disciplina autonoma non solo per i valori propri della disciplina, ma anche perchè essa offre risorse indispensabili per risolvere problemi di altre discipline: risorse operative e tecnologiche che si acquisiscono con l'addestramento; risorse metodologiche, culturali e concettuali, che si formano con lo studio e si accumulano con l'esperienza. L'obiettivo del progetto è sostituire il punto interrogativo nell'iceberg con la sigla CT, ovvero Computational Thinking. I risultati dell'analisi svolta sui risultati accumulati in due anni di esperienza sembrano indicare che la direzione in cui si muove il progetto sia soddisfacente. Al termine di questo lavoro verranno illustrati alcuni suggerimenti per rendere più incisivo e puntuale il lavoro di insegnanti e studenti.

Tabella 1.1: Contributi culturali delle discipline

Disciplina	Parte emersa	Parte sommersa
Italiano	Leggere e scrivere	Letteratura, ...
Matematica	Far di conto	Teoremi, dimostrazioni, metodo, ...
Informatica	TIC	?

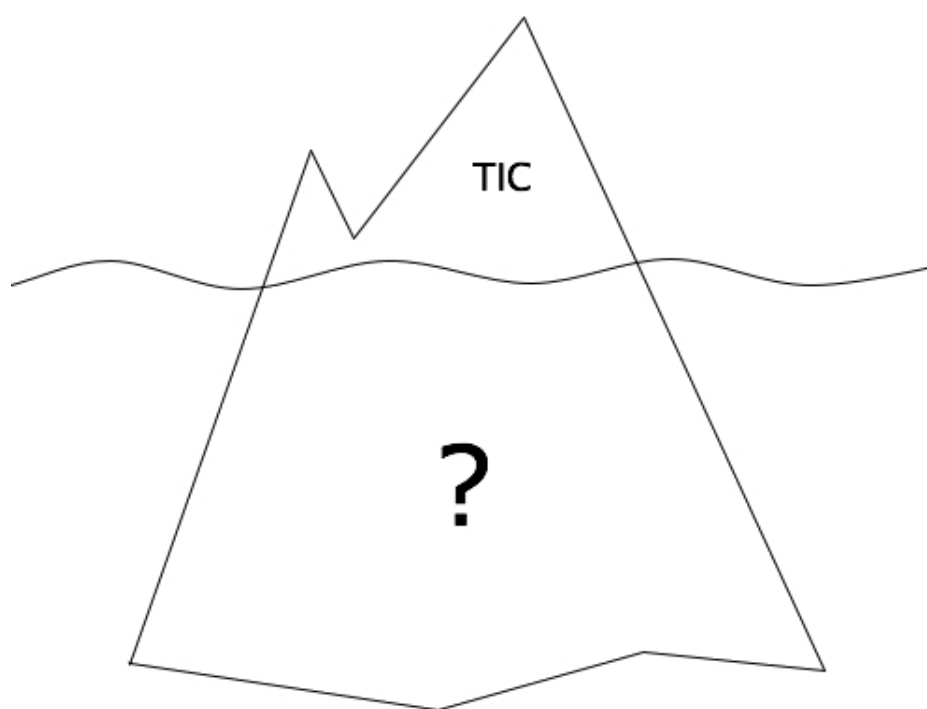


Figura 1.1: Metafora dell'iceberg

1.2 Struttura della tesi

Il Capitolo 2 presenta un'overview di quello che è il neologismo che più va di moda nella ricerca sull'educazione di questi ultimi anni, il *computational thinking*; se ne daranno definizioni e proposte di applicazione. Nel Capitolo 3 viene illustrato il progetto Olimpiadi di Problem Solving, organizzato dal MIUR, come iniziativa per diffondere il computational thinking nella scuola dell'obbligo. Nel Capitolo 4, al fine di studiare dettagliatamente tale progetto, è descritta la raccolta dei test delle olimpiadi svolti dagli studenti delle scuole partecipanti, le loro valutazioni e una personale interpretazione dei risultati. Sarà inoltre proposto un minimale adeguamento dell'impianto attuale alle necessità imposte dall'analisi statistica. Nel capitolo conclusivo sarà brevemente illustrata un'alternativa alla strutturazione attuale di contenuti e modelli di insegnamento: una vera e propria inversione di tendenza - negli ultimi anni in via di sperimentazione negli Usa - che privilegia l'aspetto attrattivo/motivazionale a quello puramente formativo. Si terminerà con le conclusioni.

Capitolo 2

Computational thinking

Il concetto di Computational Thinking si va evolvendo da qualche anno [1]. Dai primi tentativi ad oggi, data la complessità della materia, i ricercatori hanno proposto delle definizioni via via più precise ed in parte concordanti. Cercando di offrire un quadro generale completo seguirà la discussione delle definizioni che disegnano quello che è, a nostro parere, lo stato dell'arte del computational thinking oggi.

2.1 Definizione corrente

Si può pensare il *computational thinking* come un insieme ricco e in continuo sviluppo di abilità cognitive, concetti e tecniche dell'area informatica utili per il problem solving. Tuttavia non è affatto semplice fornirne una definizione rigorosa. Esso comprende il pensiero algoritmico, ma è più di questo, il ragionamento in parallelo, ma è più di questo, parole come dati, stato, comportamento, interazione, progettazione, ma è più di questo. La grande difficoltà nel catturare la sua natura è un valido motivo della necessità di affrontarlo, sia pure in modo imperfetto, incompleto e provvisorio.

Il computational thinking sfrutta la potenza e i limiti dei processi di elaborazione, siano essi eseguiti dall'uomo o da una macchina. I metodi e i modelli computazionali ci permettono di risolvere problemi e progettare sistemi che nessuno di noi sarebbe in grado di affrontare da solo. Il computational thinking ci

pone di fronte all'enigma dell'intelligenza delle macchine: che cosa l'uomo riesce a fare meglio del computer? che cosa il computer riesce a fare meglio dell'uomo?

“Computers are incredibly fast, accurate, and stupid. Human beings are incredibly slow, inaccurate, and brilliant. Together they are powerful beyond imagination” ¹

Questa citazione cattura l'essenza del computational thinking: integrare la creatività del pensiero con l'efficienza del computer. Soffermendosi su quest'ultimo concetto, ci si può chiedere in che modo si riesca a pensare in modo “computazionale”. Per rispondere è opportuno a sua volta porsi domande del tipo

- Quali sono potenza e limiti dell'intelligenza dell'uomo e del computer?
- Dato un problema, quanto, questo, è difficile da affrontare?
- In che modo può essere risolto?
- In che modo la tecnologia può essere applicata al problema?
- Quali strategie computazionali potrebbero essere impiegate?

L'idea di base nel computational thinking è quella di sviluppare dei modelli e simulare dei problemi.

Descrizioni più recenti provengono dal dipartimento di informatica della Carnegie Mellon University, coordinato da Janette M. Wing, che per prima coniò il termine “computational thinking” nel 2006. [1]

“Computational thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent.” [2]

¹Tale citazione è riportata in molti documenti scientifici, e viene solitamente attribuita ad Albert Einstein. Abbiamo altresì scoperto l'esistenza di intere documentazioni in cui scrupolosi appassionati avrebbero provato la non appartenenza di questa citazione allo scienziato. Noi ci limiteremo ad affermare che, comunque, si tratti convinzione comune.

Con computational thinking s'intende risolvere problemi, progettare sistemi e capire il comportamento umano che ricorre a concetti che nell'informatica sono fondamentali². Include un range di tool mentali che riflette la vastità del campo dell'informatica. Significa saper trattare diversi livelli di astrazione, per capire e trovare soluzioni in maniera più efficace. Significa pensare in modo algoritmico applicando concetti matematici come l'induzione per sviluppare soluzioni più sicure, efficienti ed eleganti. Significa avere familiarità con il ragionamento su larga scala, non solo per una questione di efficienza, ma anche per ragioni economiche e sociali. Il computational thinking è una specie di "pensare analitico". Condivide con il "pensare matematico" le usuali tecniche d'approccio alla risoluzione dei problemi essenzialmente basate sulla pura teoria; con il "pensare ingegneristico" i modi di progettare e valutare sistemi complessi su larga scala tenendo conto dei vincoli che il mondo reale inevitabilmente impone; con il "pensare scientifico" in generale i modi di studiare la computabilità, l'intelligenza e il comportamento della mente e dell'uomo.[3]

Il processo più significativo nel computational thinking è l'*astrazione* [2]. Conformemente a [4], interpretare e schematizzare una particolare situazione significa trovare un matching fra gli elementi della situazione e quelli di un modello di una conoscenza più generale. L'astrazione è usata per definire dei pattern, per generalizzare da specifiche istanze, per parametrizzare. È usata per permettere ad un oggetto di rappresentarne molti. È usata per catturare le proprietà essenziali comuni a un insieme di oggetti nascondendone le differenze trascurabili. Per esempio, un algoritmo è l'astrazione di un processo che prende un input, esegue una sequenza (ordinata) di passi, e produce un output che soddisfa un determinato scopo. Un tipo di dato astratto definisce un insieme astratto di valori e operazioni per manipolare questi valori, nascondendo l'effettiva rappresentazione al suo utilizzatore. L'astrazione ci permette di scalare e quindi di affrontare la complessità.

Applicando ricorsivamente l'astrazione si possono costruire sistemi sempre più

²Si sottolinea un'interpretazione in senso lato delle parole "problema" e "sistema". Infatti non si intendono soltanto problemi matematicamente ben definiti, le cui soluzioni sono completamente analizzabili - come per esempio una dimostrazione, ma anche problemi che appartengono al mondo reale, per i quali siano ottenibili solo soluzioni approssimate.

estesi e complessi. In informatica è pratica frequente costruire i software in termini di *livelli di astrazione*, permettendo di concentrarsi ogni volta su uno specifico livello e sulle relazioni con i livelli adiacenti. Le astrazioni sono il tool mentale del calcolo, di conseguenza attraverso il calcolo riusciamo ad *automatizzare le astrazioni*³. Per esempio, quando si scrive un programma in un dato linguaggio ad alto livello, si possono serenamente ignorare i dettagli dell'hardware sottostante, il sistema operativo, il file system, e/o la rete; inoltre si fa affidamento sul fatto che il compilatore implementi correttamente la semantica del linguaggio. Ma come si può misurare l'astrazione? Quali fattori la rendono "buona"? Si definisce buona un'astrazione che porta allo sviluppo di un sistema efficiente - in termini di velocità e risorse e potenza necessarie; corretto - prima o poi fornirà una risposta, e la risposta sarà quella giusta; e con una serie di caratteristiche quali semplicità, user-friendliness, modificabilità, manutenzione, costi, etc.

Come accenato in precedenza, il computational thinking ricorre sia al pensiero matematico che a quello ingegneristico. Diversamente dalla matematica i nostri sistemi di calcolo sono vincolati dalla struttura fisica dell'hardware che processa l'informazione e dal suo ambiente operativo. Per questo motivo l'informatico deve fare i conti con i casi limite (o casi particolari, cosiddetti boundary conditions), i guasti, gli agenti malevoli e tutte le imprevedibilità del mondo reale. Tuttavia, diversamente da altre discipline ingegneristiche, in informatica rimane possibile costruirsi un "mondo virtuale" distaccato e indipendente dalla realtà fisica, dove la creatività viene limitata soltanto dalla propria immaginazione. [2]

Applicazione alle altre discipline

Il computational thinking è stato utilizzato in ambito informatico, nel contesto dello sviluppo di applicazioni, per decenni. È stato anche applicato per anni in campi diversi dall'informatica. Per esempio, informatici, psicologi, sociologi, antropologi, e biologi hanno tutti contribuito ad applicare concetti e processi computazionali nel campo delle scienze cognitive [5]. L'applicazione del computational thinking all'informatica e alle discipline correlate è stata implicita, perchè esso è l'approccio naturale al problem solving. L'applicazione del computational

³The power of our "mental" tools is amplified by the power of our "metal" tools [Cit. Wing].

thinking ad altri campi è anch'essa stata solitamente implicita, talvolta senza un esplicito riconoscimento delle capacità (o abilità) di ragionamento coinvolte. Ciò che cambia rispetto alla recente attenzione per il computational thinking è l'enfasi sul definire esplicitamente che cosa è, e il suo esplicito utilizzo per il raggiungimento o la verifica di intuizioni riguardanti problemi appartenenti a campi che, apparentemente, poco o nulla hanno a che vedere con l'informatica. Come la Wing argomenta nel suo articolo seminale [1], "le idee di astrazione, livelli di astrazione, e automatizzazione, per nominarne alcuni, sono concetti di informatica fondamentali che hanno già prodotto nuove intuizioni e scoperte nel campo delle scienze naturali e delle cosiddette hard social sciences come l'economia". In totale accordo con il suo pensiero, si può sostenere che il computational thinking sia un'abilità di base emergente che dovrebbe divenire parte integrante dei processi educativi:

“Computational thinking is a fundamental skill for everyone, not just for computer scientists. To reading, writing and arithmetic, we should add computational thinking to every child’s analytical ability.”
[Jeannette M. Wing]

La maggior parte dei documenti scientifici che raccontano del computational thinking riportano molti esempi presi dalla fisica, dalla chimica e dalla biologia. Sfortunatamente però non è riportato alcun esempio che non appartenga a questa categoria, le hard sciences. Parte del lavoro di Perkovic e Settle [6] è volto all'esplorazione di esempi di computational thinking in termini più generali e in tutte le discipline accademiche. Un modo per stabilire se un processo analitico sia *computational thinking* è se questo è, o possa essere, descritto usando keyword appropriate, e se possa essere classificato come implementante un fondamentale principio di calcolo. Servendosi della terminologia riportata sotto possiamo definire delle keywords e dei principi di computazione utilizzabili per il processo di riconoscimento.

Terminologia utilizzata (definizioni naive) Per riuscire al meglio in questa classificazione sarebbe opportuno avere un'adeguata familiarità con le definizioni rigorose della seguente terminologia, affinché si crei un matching coerente fra il

linguaggio dell'informatica e quello delle altre discipline. Tuttavia siamo coscienti che è difficile che ciò accada. In letteratura [6] si ritiene che per introdurre il computational thinking nelle scuole sia necessario proporre un'idea dei concetti chiave dell'informatica come qui riportato da queste “definizioni” naive⁴:

- *Algoritmo* (o *procedura*, *funzione*): Un algoritmo è un insieme di regole che descrive come risolvere un problema. Un algoritmo può essere descritto come un programma, pseudo-codice, o una meno formale spiegazione passo-passo (volendo anche una ricetta).
- *Dato* (o *variabile*, *database*, *coda*): Il dato è un'informazione che è parte del problema, rivela come questa informazione è organizzata e acceduta. La distanza fra due città vicine è un dato per il problema del calcolo delle distanze fra le città, per fare un esempio.
- *Astrazione* (o *concettualizzazione*, *modularizzazione*): L'astrazione è l'estrapolazione di proprietà importanti e la generalizzazione delle relazioni.
- *Oggetto*: Un'entità che ha certe proprietà e può eseguire certe azioni. Un essere umano, un'automobile, o un'applicazione di un calcolatore sono tutti oggetti.
- *Processo*: L'esecuzione di un algoritmo. Un processo potrebbe essere un essere umano, una comunità, o un programma in esecuzione che sta eseguendo qualche task.
- *Sistema*: Un sistema è un gruppo di processi o oggetti che interagiscono. Un sistema potrebbe essere una rete di computer, uno stormo di uccelli, una comunità virtuale, così come entità più vaste come sistemi economici e biologici.
- *Iterazione* (o *loop*, *ricorsione*): L'iterazione riguarda la ripetizione di una o più operazioni fino al raggiungimento di una condizione desiderata. Nella progettazione e sviluppo di un gioco, un esempio di iterazione si presenta

⁴Le keywords possono essere accompagnate da alcuni sinonimi utili per un matching più significativo

nella costruzione delle versioni del gioco, dal prototipo iniziale alla versione beta.

Principi di computazione e keywords. Computazione è un termine abbastanza generale che comprende diversi task, concetti e tecniche. Analogamente, computational thinking coinvolge un'ampia insieme di approcci e abilità. Per capire il computational thinking distinguendo differenze e somiglianze fra esempi specifici in campi diversi, è utile definire diverse categorie di computational thinking. Noi faremo riferimento alle categorie definite da Denning [7]. Lo scopo del suo progetto è di distinguere i fondamentali principi della computazione. È particolarmente interessante in questo contesto una delle sue motivazioni: “Per stabilire una nuova relazione con gente proveniente da altri campi, offrendo loro dei principi di computazione che possano essere facilmente mappati nel loro contesto di lavoro”. Egli afferma che “i principi di computazione possono essere organizzati in sette categorie, ognuna delle quali enfatizzi una visione univoca sulla computazione”. In accordo con Denning distinguiamo quindi *computazione*, *comunicazione*, *coordinazione*, *memoria*, *automatizzazione*, *valutazione*, e *design*. Questi sette principi formano una solida base utile per riconoscere, organizzare e classificare le istanze di computational thinking, e costituiscono un framework capace di tradurre il computational thinking in contesti diversi dall'informatica. Naturalmente le categorie devono essere viste come un punto di partenza, soggetto a modifiche quando necessario. Le definizioni che noi prendiamo in considerazione, per essere il più possibile generali, sono quelle proposte da Perkovic e Settle [Perkovic10]. Per ogni principio sono anche elencate delle keyword, tipicamente utilizzate per esprimere concetti che il principio stesso esprime.

Computazione. È l'esecuzione di un algoritmo, un processo che inizia da uno stato iniziale contenente l'algoritmo e i dati di input, e passa attraverso un numero indeterminato di stati fino a quello finale, che determina il raggiungimento dello scopo.

Keywords: *stato e transizione di stato, algoritmo, programma, ricerca esaustiva, backtracking, ricorsione e iterazione, albero di decisione, randomizzazione, complessità, calcolabilità.*

Comunicazione È la trasmissione di informazioni da un processo (o oggetto) a un altro.

Keywords: *informazione e sua rappresentazione, messaggio, mittente/destinatario, protocollo di comunicazione, compressione di un messaggio, cifratura di un messaggio, correzione d'errore, canale di comunicazione, encoder/decoder, rumore, autenticazione.*

Coordinazione È il controllo (attraverso la comunicazione, per esempio) del tempo di computazione dei processi partecipanti al fine di ottenere lo scopo desiderato.

Keywords: *processi/agenti interagenti, protocolli inter-processo che includono protocolli di comunicazione, sincronizzazione, eventi e gestione degli eventi, dipendenza, concorrenza.*

Memoria È la codifica e l'organizzazione dei dati un modo da rendere efficienti le ricerche e le altre operazioni.

Keywords: *immagazzinamento, gerarchia e organizzazione dei dati, manipolazione dei dati come p.e. inserimenti/accodamento/rimozione, localizzazione dei dati e caching, rappresentazione virtuale, sistemi di naming, riferimenti assoluti e relativi.*

Automatizzazione È il mapping delle computazioni nei sistemi fisici che le eseguono.

Keywords: *mapping di un algoritmo in un oggetto di una computazione fisica, meccanizzazione, si applica a processi ripetitivi e offre esecuzioni consistenti, senza errori, veloci e poco costose computazionalmente.*

Valutazione È l'analisi statistica, numerica e sperimentale dei dati.

Keywords: *visualizzazione, analisi dei dati, statistica, data mining, simulazione, esperimento computazionale.*

Design È l'organizzazione (usando astrazione, modularizzazione, aggregazione, scomposizione) di un sistema, di un processo, di un oggetto, etc.

Keywords: *astrazione, livelli di astrazione, modellazione, modularità, occultazione dell'informazione, classe, architettura, aggregazione, pattern, struttura sottostante.*

2.2 Computational thinking nella didattica

Google ha intrapreso la missione di promuovere il computational thinking nelle scuole fino al cosiddetto curriculum K-12 - che in Italia corrisponde al percorso di studi che va dalla scuola primaria alla fine alla scuola secondaria di secondo grado - per assistere gli studenti nell'apprendimento e svelare a tutti l'abilità del XXI secolo. [8]

Viene da chiedersi, a questo punto, perchè proprio Google sia interessato all'insegnamento del computational thinking ai giovani studenti. Il computational thinking coinvolge un insieme di abilità e tecniche di problem solving che gli ingegneri del software utilizzano per costruire i programmi che sono alla base delle applicazioni che ognuno di noi, ogni giorno, utilizza - posta elettronica, motori di ricerca e mappe per portare degli esempi. Tecniche specifiche di computational thinking includono:

- **Decomposizione:** L'abilità di suddividere un problema in più sottoproblemi di dimensione ridotta (talvolta irriducibili), così da poter spiegare chiaramente un procedimento ad un'altra persona o a un computer, o anche soltanto prendere nota per noi stessi. Decomporre un problema porta di solito al riconoscimento di pattern e alla generalizzazione, e dunque anche alla capacità di progettare un algoritmo.

Esempio: Quando assaggiamo un piatto nuovo e cerchiamo di identificare i diversi ingredienti in base al loro sapore, lo stiamo decomponendo nei suoi singoli ingredienti.

- **Riconoscimento di pattern:** L'abilità di intravedere somiglianze o differenze comuni che aiuteranno a fare predizioni o porteranno a scorciatoie. Il riconoscimento di pattern è solitamente alla base del problem solving e della progettazione di algoritmi.

Esempio: I bambini identificano pattern di comportamento nella reazione dei propri genitori e degli insegnanti ai propri comportamenti, per determinare cosa è giusto e cosa è sbagliato. Proprio questi pattern condizioneranno i loro comportamenti futuri.

- **Generalizzazione di pattern e astrazione:** L'abilità di filtrare l'informazione non necessaria per la risoluzione di un certo tipo di problema e generalizzare invece l'informazione necessaria. La generalizzazione di pattern e l'astrazione permettono di rappresentare un'idea o un processo in termini generali, affinché li si possa utilizzare per risolvere altri problemi della stessa natura.

Esempio: In matematica possiamo generalizzare le formule usando delle variabili⁵ invece dei numeri, così da poterle utilizzare per risolvere problemi che coinvolgono valori sempre diversi.

- **Progettazione di algoritmi:** L'abilità nello sviluppare una soluzione passo-passo per un dato problema. La progettazione di algoritmi è spesso basata sulla decomposizione e sull'identificazione di pattern che aiutino la risoluzione di un problema. In informatica, così come in matematica, gli algoritmi sono spesso scritti in modo astratto, cioè descritti, utilizzando delle variabili al posto di numeri specifici.

Esempio: Quando uno chef scrive la ricetta di un piatto sta semplicemente progettando e creando un algoritmo che altri potranno seguire nel cercare di replicare (più o meno fedelmente!) quel piatto.

Data la crescente prevasività della tecnologia nella vita di tutti i giorni e in moltissime carriere lavorative al di fuori dell'ambito informatico, Google vuole supportare la costruzione e lo sviluppo di queste abilità di problem solving per aiutare ad innalzare il livello base di comprensione di ogni individuo.

2.3 Computational thinking e altre discipline

Il computational thinking ha già influenzato i piani di ricerca di tutte le discipline scientifiche e ingegneristiche. Iniziando qualche decina d'anni fa con i modelli computazionali e le simulazioni per lo studio di sistemi complessi su larga scala, e proseguendo attraverso il più recente utilizzo di data mining e machine learning per l'analisi di enormi moli di dati, la *computazione* è stata

⁵Come ad esempio nella formula $-b \pm \frac{\sqrt{b^2 - 4ac}}{2a}$

riconosciuta come il terzo pilastro della conoscenza scientifica assieme a *teoria* e *sperimentazione* [9]

Quando si cerca di aiutare qualcuno - in questo caso qualcuno che sta al di fuori dell'ambito informatico - è importante sottolineare i possibili benefici dell'applicare il computational thinking. Esiste una tendenza, probabilmente giustificabile, ad essere scettici nell'applicazione di nuovi approcci, specialmente quando quelli esistenti danno e hanno sempre dato risultati soddisfacenti, e quando questi nuovi approcci provengono da aree che non sono quelle alle quali essi stessi sono rivolti.

Per illustrare questi concetti al mondo della scuola, spesso caratterizzato da eccessivo conservatorismo, può essere utile elencare alcuni esempi. Nonostante l'impatto della computazione sia stato riconosciuto da poco tempo (il premio Nobel per la fisica Kenneth G. Wilson ne riconobbe il ruolo nella sua Nobel lecture, nel 1982 [10]), oggi la fisica dei sistemi complessi come, ad esempio, il comportamento di uno stormo di uccelli [11] è meglio compresa attraverso processi computazionali maggiormente inclini ad analisi mediante l'uso di simulazioni e tool per la complessità computazionale. Il computational thinking ha pervaso il campo dell'economia, dalla finanza computazionale all'eTrading: sta generando un nuovo campo di microeconomia computazionale, con applicazioni come ad placement, aste online, sistemi di reputazione e modelli di scambio di donatori di reni [12]. Potendo archiviare una quantità pressochè illimitata di documenti nei database, si è potuto cambiare il modo di fare ricerca. Si sta trasformando la statistica, dove, con il machine learning, l'automatizzazione dei metodi bayesiani e l'uso di modelli grafici probabilistici rendono possibile identificare pattern e anomalie in datasets di grosse dimensioni e dai contenuti più disparati (mappe astronomiche, test MRI⁶, acquisti con carte di credito e ricevute del droghiere, ...) [13]. Per quanto riguarda la matematica, la scoperta del Gruppo di Lie E_8 ha reso necessaria una collaborazione fra 18 matematici e informatici e l'aiuto di un supercomputer. La scoperta di tale gruppo ha avuto un pesante impatto sulla fisica teorica, in particolare le teorie delle stringhe e della supergravità. Sempre in ambito matematico, la dimostrazione del Teorema dei Quattro Colori (1977) fu una delle prime dimostrazioni eseguite usando un computer. Il sequenziamento velocizzato del genoma umano attraverso l'algoritmo *shotgun*, prima, e le simu-

⁶Magnetic resonance imaging

lazioni dei processi dinamici presenti in natura - il ciclo di divisione cellulare, il ripiegamento delle proteine, ... [14], poi, hanno destato l'interesse dei biologi per i metodi computazionali, non soltanto per i mezzi (computers e reti). I musicisti e molti altri artisti hanno imparato a intendere il calcolo come un modo di accrescere la loro creatività e dunque anche la loro produttività. In altri campi il computational thinking si trova ancora ad uno stadio elementare: giorni e giorni di simulazioni di modelli matematici di processi fisici trovati in natura.

2.4 Il computational thinking nelle scuole

Un numero considerevole di campus universitari americani - e non solo, sebbene da lì si sia partiti - sta rivisitando il curriculum della laurea in informatica. Molti stanno riformando i corsi di informatica di primo livello, per trasmetterne i principi e i concetti fondamentali (non soltanto le tecniche di programmazione). La facoltà di informatica della Carnegie Mellon University ha apportato delle modifiche strutturali al proprio corso introduttivo, affinché un certo tipo di abilità possa essere percepito non soltanto dagli studenti, appunto, di informatica, ma anche dai partecipanti provenienti dal resto del campus. Le principali modifiche sono tre: 1) rivisitazione del corso introduttivo per promuovere i principi del computational thinking, per majors⁷ e nonmajors, 2) sottolineare l'enfasi sulla necessità di rendere altamente affidabili i sistemi software e sui mezzi per ottenere ciò, e 3) preparare gli studenti ad un futuro in cui i programmi raggiungeranno un alto livello di performances con l'esecuzione parallela. [15]

L'interesse e l'entusiasmo attorno al computational thinking non si sono sviluppati soltanto nell'ambiente universitario. La possibilità di diffondere il computational thinking attraverso la popolazione ha infatti motivato diversi progetti, di qui a seguire si riporta una lista non esaustiva.

- L'associazione College Board, col supporto della NSF⁸, ha fatto partire Advanced Placement pilota - *Computer Science: Principles* - con l'obiettivo di

⁷Il livello cosiddetto *major* nel sistema scolastico americano rappresenta una categoria di corsi da seguire per ottenere il titolo di studio - di livello *under graduate* - chiamato *Bachelor's degree*.

⁸National Science Foundation

presentare i concetti fondamentali del calcolo e del computational thinking (<http://csprinciples.org>). Giunti nel 2011-12 al secondo anno di sperimentazione, si registrano nove high schools partecipanti, aventi come referenti altrettante università: Georgia Tech, Illinois Institute of Technology, Trinity College, University of Alabama, University of Arkansas at Little Rock, University of Pennsylvania, University of Wisconsin-Madison, Virginia Tech e Stanford University. Periodo: anno 2010.

- La Computer Science and Telecommunications Board propone una serie di workshops sul tema “Computational thinking per tutti” cercando in particolar modo di identificare quali principi di informatica è opportuno insegnare agli studenti liceali. [16]
- The Hill, quotidiano congressuale, pubblicizza un evento sul tema “putting the “C” into STEM⁹”. Sponsor dell’evento sono Association for Computing Machinery, Computing Research Association, Computer Science Teachers Association, IEEE, Microsoft, National Center for Women and Information Technology, Society for Women Engineers, National Science Foundation. Periodo: maggio 2009.
- La Camera dei Rappresentanti degli Stati Uniti d’America istituisce la “Computer Science Education Week” (www.csedweek.org). Sponsor dell’evento sono Anita Borg Institute for Women and Technology, Association for Computing Machinery, Business-Higher Education Forum, Computing Research Association, Computer Science Teachers Association, Dot Diva, Google, Globaloria, Intel, Microsoft, National Center for Women and Information Technology, National Science Foundation, SAS e Upsilon Pi Epsilon. Periodo: settimana che contiene il 9 dicembre (in onore di Grace Hopper).
- Proposta di legge (H.R.5929, Computer Science Education Act) per rafforzare l’educazione informatica nella scuola primaria e secondaria. Periodo: luglio 2010.

⁹L’acronimo STEM sta per scienze, tecnologia, ingegneria e matematica, e sta ad intendere tutte quelle accademie e discipline professionali che ricadono in quest’area.

- Dalla NSF parte il programma “Computing Education for the 21st Century (CE21, <http://www.nsf.gov/pubs/2010/nsf10619/nsf10619.htm>) per trasmettere le competenze di computational thinking agli studenti di livello K-14 (ovvero dei primi due anni di università) e agli insegnanti. CE21 sfrutta il successo di due progetti precedenti, NSF CISE¹⁰ “Pathways to Revitalized Undergraduate Computing Education” (CPATH, <http://www.nsf.gov/>) e “Broadening Participating in Computing” (BPC). CE21 dedica particolare enfasi alle attività che supportano il progetto “CS/10K”¹¹. Periodo: settembre 2010.
- La British Royal Society annuncia di essere a capo della condotta di un progetto della durata di 18 mesi per la verifica dei modi in cui la computazione viene insegnata nelle scuole. Il progetto si avvale della collaborazione di 24 organizzazioni provenienti da tutta la comunità del settore, incluse società culturali, ordini professionali, università, e industrie (<http://royalsociety.org/education/policy/>). Periodo: agosto 2010.
- Dal 2006 CS4HS (Computer Science for High School, <http://www.cs4hs.com/>) è un’iniziativa sponsorizzata da Google per promuovere l’informatica e il computational thinking nelle scuole medie e superiori. Con la concessione di un Education Group di Google, le università organizzano dei workshops della durata di due o tre giorni per gli insegnanti di informatica delle vicine scuole medie/superiori, per far ottenere loro una sempre più consolidata consapevolezza che *informatica* voglia dire molto di più rispetto a come viene percepita la sigla TIC o anche alla sola *programmazione*.
- Dal 2007 Microsoft Research ha fondato il Carnegie Mellon Center for Computational Thinking (<http://www.cs.cmu.edu/CompThink/>). Il progetto supporta sia i progetti di ricerca che quelli di sensibilizzazione all’educazione.

¹⁰Computer and Information Science and Engineering

¹¹Dal directorato CISE della NSF è emersa una proposta di revisione del curriculum sulla formazione informatica nelle scuole superiori attraverso tale progetto. L’intento del CS/10K Project è sviluppare un curriculum effettivo applicato in 10000 scuole con la collaborazione di 10000 insegnanti, entro il 2015.

- Google lancia il sito web “Exploring Computational Thinking”. Al suo interno si possono trovare ulteriori link a risorse web, incluso un piano di lezioni modello per gli insegnanti, suddiviso per tipologia e anno di scuola (da K-4 a K-12). Periodo: ottobre 2010. [8]
- Computer Science Unplugged (<http://csunplugged.org/>), creato da Tim Bell, Mike Fellows e Ian Witten, insegna l’informatica senza l’uso del computer. Si tratta di una raccolta di attività di apprendimento che permette di imparare l’informatica attraverso giochi e puzzles che usano carte, stringhe, matite, e un po’ di movimento. Le attività introducono gli studenti a concetti fondamentali come numero binario, algoritmo e compressione di un dato. CS Unplugged è adatto a persone di tutte le età.
- White paper presentato dalla CRA-E¹² che include suggerimenti su corsi di computational thinking per studenti universitari.
- Recente impegno del Regno Unito nel continuo intento di rendere più corpositi i propri corsi di alfabetizzazione informatica.
- Singapore: si cerca di incorporare il computational thinking nel curriculum K-12, tuttavia ad oggi non sono ancora stati concretizzati i rinnovamenti in progetto. (guardare al “CS Reloaded Programme”)
- Corsi a tutti i livelli in Hong Kong e Cina.

¹²Computing Research Association - Education

Capitolo 3

Progetto problem solving

3.1 Introduzione

In molti ritengono che il computational thinking possa essere visto come un nuovo termine per riferirsi a un concetto chiave, il pensiero algoritmico [17]. In realtà, come precedentemente discusso, i termini come pensiero algoritmico, pensiero matematico, pensiero ingegneristico, problem solving, etc. sono fortemente intersecati fra loro tanto da rendere difficile una vera e propria definizione dei loro limiti. Recentemente ci si riferisce all'insieme di questi concetti col neologismo *computational thinking*. Tutte queste discussioni hanno portato, l'informatica, più o meno esplicitamente, ad essere considerata maggiormente, dati i cambiamenti nei processi educativi indotti dalla diffusa pervasività delle tecnologie nell'informazione e nella comunicazione. Tuttavia la percezione che la maggior parte delle persone ha dell'informatica - insegnanti compresi - è sbagliata.

“Computer science is no more about computers than astronomy is about telescope.” [E. W. Dijkstra]

L'informatica non è soltanto un computer, o i linguaggi di programmazione, o un insieme di servizi. Va concepita come modello concettuale (computational thinking) che necessita dell'utilizzo di un suo linguaggio (computer programming) - in realtà non esiste soltanto un linguaggio, ma una vera e propria Babele dei linguaggi di programmazione [18] - per formalizzare e risolvere problemi di ogni disciplina. Al MIUR, Ministero dell'Istruzione, dell'Università e della Ricerca,

si è ritenuto necessario, con l'istituzione del progetto "Olimpiadi di problem solving", far conoscere l'informatica come disciplina scientifica trasversale utile a rendere più ricco, per gli studenti (e non solo), l'insieme di conoscenze, abilità e competenze in vista del loro futuro inserimento nella società.

3.2 Specifiche

La proposta di una competizione di problem solving rivolta all'intero ciclo della scuola dell'obbligo persegue una doppia finalità:

- Sfruttare la potenziale pervasività applicativa della metodologia del problem solving
- Avviare e consolidare una visione concettuale informatica, quindi non solo tecnologica, negli alunni sin dai primi anni di formazione, mobilitando processi e prodotti che sottolineano la connotazione di disciplina scientifica, fruibile come "metodo concettuale che consente di formalizzare e risolvere problemi in ogni campo".

Intersecando informatica e problem solving si sollecitano azioni cognitive in cui si rilancia il primato del processo sul prodotto e in cui le conoscenze procedurali e le conoscenze dichiarative si coniugano per la gestione costruttiva delle diverse dimensioni formative sovradisciplinari e trasversali alle discipline, ma che trovano fondamento nelle conoscenze/abilità disciplinari. Infatti, le prove proposte sono radicate nelle aree disciplinari di base, ma sono orientate a stimolare percorsi di ricerca in cui giocano d'assalto le competenze proprie del problem solving: ricerca, esplorazione ed analisi di tutti i dati, necessari-superflui-alternativi, da organizzare per trovare e rappresentare percorsi di risoluzione attraverso format di sintesi logica. In quanto metodologia il problem solving rimanda ad attività in cui prevale il pensare, il ragionare, il fare ipotesi ed operare scelte, attività che richiedono l'applicazione di abilità relative alla gestione di informazioni strutturali più che l'applicazione sterile di procedimenti meccanici volti alla risoluzione di semplici calcoli. Attività che valorizzano l'instaurarsi di quelle competenze trasversali ai diversi contesti disciplinari riconosciute ormai essenziali per un inserimento attivo e consapevole dei giovani nella società. Le abilità di problem

solving si apprendono esercitandole sistematicamente in tutti i contesti di vita scolastica ed extrascolastica, per cui tutte le prove proposte vanno assunte soprattutto come un programma aperto. Il supporto delle linee-guida come indicazione metodologica, del syllabus come insieme di indicazioni contenutistiche, consentirà ai docenti di muoversi in modo disinvolto nelle situazioni problematiche secondo i principi cardine sia del linguaggio informatico sia del problem solving.

3.3 Syllabus

Premessa La pervasività dell'informatica e il suo essere indispensabile nella vita quotidiana hanno reso necessario l'inserimento del suo insegnamento nei processi formativi. Di tale esigenza, presente sia nelle Indicazioni Nazionali (2004), sia nelle Indicazioni per il curricolo (2007), si è resa conto la parte più sensibile e attenta della scuola italiana che, da tempo e a vario titolo, ha introdotto le tecnologie informatiche nell'attività curricolare o extracurricolare. L'Informatica va insegnata, studiata e capita non tanto per formare bravi professionisti della disciplina, quanto perché la conoscenza dei suoi fondamenti contribuisce a formare e arricchire il bagaglio tecnico, scientifico e culturale di ogni persona. Essa, infatti, assume un duplice ruolo nell'insegnamento: da una parte ruolo culturale e formativo di disciplina scientifica di base (a fianco della matematica e delle scienze); dall'altra ruolo di strumento concettuale trasversale a tutte le discipline.

Nella scuola dell'obbligo, quindi, l'Informatica deve essere introdotta per le seguenti motivazioni culturali:

1. La conoscenza dei contenuti fondamentali (syllabus) è substrato indispensabile per creare le premesse dell'utilizzo della logica nelle attività di organizzazione della conoscenza e nella costruzione delle competenze;
2. La conoscenza di metodologie e tecniche di base della programmazione, dell'algoritmica e della rappresentazione dei dati è una risorsa concettuale particolarmente adatta per acquisire e saper usare competenze e abilità generali di problem solving;

3. Le abilità strumentali che consentono di usare i servizi offerti da Internet e dai cosiddetti software didattici, disponibili per ogni disciplina del curriculum, sono ormai un presupposto essenziale per le scuole di ogni ordine e grado

Per questi motivi il syllabus è suddiviso in due componenti: elementi di informatica e strumenti. La prima colonna rappresenta l'informatica come scienza e metodologia, che fonda e dà un supporto linguistico al problem solving. La seconda rappresenta gli strumenti e le applicazioni dell'informatica che sono essenziali per una cittadinanza responsabile nella società dell'informazione.

Finalità Il syllabus di “Elementi di informatica” non è stato stilato con l'intenzione di tracciare un quadro dei grandi concetti fondanti e portanti dell'informatica ma, in modo molto più semplice, è il frutto dell'esperienza della didattica nella scuola. Ne risulta un profilo concettuale di nozioni di base importanti e irriducibili; è l'informatica che costituisce il riferimento concreto, elementare ma tuttavia indispensabile, per costruire con sicurezza e senza ambiguità le competenze essenziali di logica della conoscenza. L'obiettivo che si è voluto raggiungere, dunque, non è quello di offrire un quadro culturale di strumenti informatici avanzati, ma un elenco semplice e rassicurante di irrinunciabili elementi di informatica di base, con riguardo anche agli aspetti etici, sociali e giuridici. Con riferimento a [19], la Tabella 3.1 riguarda il ciclo di istruzione primario (quarta e quinta), mentre le Tabelle 3.2 e 3.3 il ciclo di istruzione secondario di primo e secondo grado, rispettivamente.

3.4 Assi culturali

Gli assi culturali costituiscono il “tessuto” per la costruzione di percorsi di apprendimento orientati all'acquisizione delle competenze chiave che preparino i giovani alla vita adulta e che costituiscano la base per consolidare e accrescere saperi e competenze in un processo di apprendimento permanente, anche ai fini della futura vita lavorativa. Gli assi culturali che caratterizzano l'obbligo di istruzione sono quattro: dei linguaggi, matematico, scientifico-tecnologico, storico-sociale. (tratto da [20])

Secondo biennio: quarta e quinta classe	
<p>Elementi di Informatica</p> <p>Descrizione di procedimenti con pseudo codice non formalizzato;</p> <p>Formalizzazione di risoluzione di problemi molto semplici e loro specifica in un linguaggio formale usando percorsi;</p> <p>Linguaggi logici e semplici procedure informatiche;</p> <p>Alberi di decisione;</p> <p>Utilizzo in situazioni di gioco del linguaggio della probabilità;</p> <p>Algoritmi di semplici procedure (ordinamento, calcolo, ragionamento logico matematico e situazioni reali);</p> <p>Rilevazione e registrazione di dati, anche automatica;</p> <p>Rappresentazione dei dati mediante grafici e tabelle;</p> <p>Il metodo top-down e alberi di risoluzione di problemi;</p> <p>Ricerca e descrizione di percorsi in un grafo;</p> <p>Scrittura di semplici programmi;</p> <p>La nozione di ipertesto: progettazione e costruzione di semplici ipertesti;</p> <p>Cenni alla rappresentazione digitale di informazione non testuale (suono, immagine, etc.);</p> <p>Ricerca e classificazione delle informazioni</p>	<p>Strumenti</p> <p>Le componenti del computer e le periferiche in base alla funzione;</p> <p>I principali tipi di supporto digitale (CD-Rom, DVD, ...);</p> <p>Creazione e gestione di finestre e cartelle;</p> <p>Organizzazione del desktop;</p> <p>Documenti multimediali: inserimento di immagini in un testo;</p> <p>I principali strumenti di costruzione di disegni: matita, testo, gomma, pennello, colori, linee;</p> <p>Programmi di videoscrittura, inserimento di tabelle;</p> <p>Rappresentazione dei caratteri in forma binaria;</p> <p>Definizione delle nozioni di bit e di byte;</p> <p>Memorizzare dati su supporti digitali diversi;</p> <p>Utilizzo di CD-Rom e DVD;</p> <p>Regole e linee guida per l'utilizzo consapevole e corretto delle informazioni disponibili sul WWW;</p> <p>La netiquette della navigazione e della posta elettronica;</p> <p>Uso di software didattici (proprietary e open source) per approfondire contenuti disciplinari</p>

Tabella 3.1: Syllabus scuola primaria

Prima, seconda e terza classe	
<p>Elementi di Informatica</p> <p>Definizione di algoritmo e progettazione di semplici algoritmi di tipo combinatorio;</p> <p>Formalizzazione e scomposizione di problemi in sotto problemi;</p> <p>Scrittura di semplici algoritmi/programmi con l'uso di pseudocodice non formalizzato;</p> <p>Esercitazioni logiche, matematiche e geometriche;</p> <p>Le nozioni di vero e di falso e la nozione di proposizione;</p> <p>I connettivi congiunzione, disgiunzione e negazione;</p> <p>Proposizioni semplici e proposizioni composte;</p> <p>Combinazione di più connettivi, quantificatori, e combinazione con i connettivi;</p> <p>Implicazione e doppia implicazione;</p> <p>Regole di inferenza e ragionamenti;</p> <p>Collegamenti con gli insiemi;</p> <p>Modelli reali, modelli matematici e simulazioni;</p> <p>Rappresentazione di un brano letterario (o di un brano di storia) in linguaggio ipertestuale</p>	<p>Strumenti</p> <p>Uso avanzato dei programmi per la gestione di testi (tabelle, elementi grafici);</p> <p>Uso di software per presentazioni;</p> <p>Uso elementare di un foglio di calcolo; rappresentazione di dati attraverso grafici di tipo statistico;</p> <p>L'interfaccia del sistema operativo: cartelle (directory) e file, gestione dell'interfaccia grafica e dei sistemi di sicurezza;</p> <p>Uso di ambienti di ricerca web;</p> <p>Acquisizione e modifica delle immagini: descrizione delle caratteristiche di una immagine digitale;</p> <p>Conoscere le unità di misura della memoria (bit, byte, KB, MB, GB) e saperle attribuire ai principali supporti di memoria digitali e a documenti tipo (una lettera, un video, una cartella di 10 foto, etc.);</p> <p>Utilizzazione di semplici ambienti interattivi e simulazioni</p>

Tabella 3.2: Syllabus per le tre classi del ciclo di istruzione secondario di primo grado

Primo biennio	
<p>Elementi di Informatica</p> <p>Comprensione di semplici algoritmi fondamentali: di ordinamento, di visita su grafi/alberi, di cammini su grafi, di flusso su reti;</p> <p>Scrittura di programmi con l'uso di un linguaggio di programmazione, o con pseudo-codice formale;</p> <p>Uso di un linguaggio di programmazione (o pseudocodice) per suddividere un problema in sottoproblemi (funzioni e procedure);</p> <p>Il computer come gerarchia di macchine (il livello delle applicazioni, il livello del sistema operativo, il livello della macchina fisica);</p> <p>La specificità dei diversi linguaggi di programmazione;</p> <p>Gestione dell'informazione: modelli dei dati, concetti introduttivi sulle basi di dati;</p> <p>Uso responsabile del web: proprietà intellettuale, privacy e riservatezza</p>	<p>Strumenti</p> <p>Uso consapevole dei programmi di scrittura, di presentazione, di foglio elettronico;</p> <p>Uso consapevole delle applicazioni per la navigazione su web e per la collaborazione a distanza (chat, posta elettronica, web 2.0, etc.);</p> <p>Uso di ambienti interattivi e simulativi di supporto ad altre discipline (laboratorio di fisica, chimica, lettere, etc.)</p>

Tabella 3.3: Syllabus per le classi I e II del ciclo di istruzione secondario di secondo grado

L'Asse dei linguaggi L'asse dei linguaggi ha l'obiettivo di far acquisire allo studente la padronanza della lingua italiana come ricezione e come produzione, scritta e orale; la conoscenza di almeno una lingua straniera; la conoscenza e la fruizione consapevole di molteplici forme espressive non verbali; un adeguato utilizzo delle tecnologie dell'informazione e della comunicazione. La padronanza della lingua italiana è premessa indispensabile all'esercizio consapevole e critico di ogni forma di comunicazione; è comune a tutti i contesti di apprendimento ed è obiettivo delle discipline afferenti ai quattro assi. Il possesso sicuro della lingua italiana è indispensabile per esprimersi, per comprendere e avere relazioni con gli altri, per far crescere la consapevolezza di sé e della realtà, per interagire adeguatamente in una pluralità di situazioni comunicative e per esercitare pienamente la cittadinanza. Le competenze comunicative in una lingua straniera facilitano, in contesti multiculturali, la mediazione e la comprensione delle altre culture; favoriscono la mobilità e le opportunità di studio e di lavoro. Le conoscenze fondamentali delle diverse forme di espressione e del patrimonio artistico e letterario sollecitano e promuovono l'attitudine al pensiero riflessivo e creativo, la sensibilità alla tutela e alla conservazione dei beni culturali e la coscienza del loro valore. La competenza digitale arricchisce le possibilità di accesso ai saperi, consente la realizzazione di percorsi individuali di apprendimento, la comunicazione interattiva e la personale espressione creativa. L'integrazione tra i diversi linguaggi costituisce strumento fondamentale per acquisire nuove conoscenze e per interpretare la realtà in modo autonomo.

L'asse matematico L'asse matematico ha l'obiettivo di far acquisire allo studente saperi e competenze che lo pongano nelle condizioni di possedere una corretta capacità di giudizio e di sapersi orientare consapevolmente nei diversi contesti del mondo contemporaneo. La competenza matematica, che non si esaurisce nel sapere disciplinare e neppure riguarda soltanto gli ambiti operativi di riferimento, consiste nell'abilità di individuare e applicare le procedure che consentono di esprimere e affrontare situazioni problematiche attraverso linguaggi formalizzati. La competenza matematica comporta la capacità e la disponibilità a usare modelli matematici di pensiero (dialettico e algoritmico) e di rappresentazione grafica e simbolica (formule, modelli, costrutti, grafici, carte), la capacità di comprendere

ed esprimere adeguatamente informazioni qualitative e quantitative, di esplorare situazioni problematiche, di porsi e risolvere problemi, di progettare e costruire modelli di situazioni reali. Finalità dell'asse matematico è l'acquisizione, al termine dell'obbligo d'istruzione, delle abilità necessarie per applicare i principi e i processi matematici di base nel contesto quotidiano della sfera domestica e sul lavoro, nonché per seguire e vagliare la coerenza logica delle argomentazioni proprie e altrui in molteplici contesti di indagine conoscitiva e di decisione.

L'asse scientifico-tecnologico L'asse scientifico-tecnologico ha l'obiettivo di facilitare lo studente nell'esplorazione del mondo circostante, per osservarne i fenomeni e comprendere il valore della conoscenza del mondo naturale e di quello delle attività umane come parte integrante della sua formazione globale. Si tratta di un campo ampio e importante per l'acquisizione di metodi, concetti, atteggiamenti indispensabili ad interrogarsi, osservare e comprendere il mondo e a misurarsi con l'idea di molteplicità, problematicità e trasformabilità del reale. Per questo l'apprendimento centrato sull'esperienza e l'attività di laboratorio assumono particolare rilievo. L'adozione di strategie d'indagine, di procedure sperimentali e di linguaggi specifici costituisce la base di applicazione del metodo scientifico che - al di là degli ambiti che lo implicano necessariamente come protocollo operativo - ha il fine anche di valutare l'impatto sulla realtà concreta di applicazioni tecnologiche specifiche. L'apprendimento dei saperi e delle competenze avviene per ipotesi e verifiche sperimentali, raccolta di dati, valutazione della loro pertinenza ad un dato ambito, formulazione di congetture in base ad essi, costruzioni di modelli; favorisce la capacità di analizzare fenomeni complessi nelle loro componenti fisiche, chimiche, biologiche. Le competenze dell'area scientifico-tecnologica, nel contribuire a fornire la base di lettura della realtà, diventano esse stesse strumento per l'esercizio effettivo dei diritti di cittadinanza. Esse concorrono a potenziare la capacità dello studente di operare scelte consapevoli ed autonome nei molteplici contesti, individuali e collettivi, della vita reale. E' molto importante fornire strumenti per far acquisire una visione critica sulle proposte che vengono dalla comunità scientifica e tecnologica, in merito alla soluzione di problemi che riguardano ambiti codificati (fisico, chimico, biologico e naturale) e aree di conoscenze al confine tra le discipline anche diversi da quelli su

cui si è avuto conoscenza/esperienza diretta nel percorso scolastico e, in particolare, relativi ai problemi della salvaguardia della biosfera. Obiettivo determinante è, infine, rendere gli alunni consapevoli dei legami tra scienza e tecnologie, della loro correlazione con il contesto culturale e sociale con i modelli di sviluppo e con la salvaguardia dell'ambiente, nonché della corrispondenza della tecnologia a problemi concreti con soluzioni appropriate.

L'Asse storico-sociale L'asse storico-sociale si fonda su tre ambiti di riferimento: epistemologico, didattico, formativo. Le competenze relative all'area storica riguardano, di fatto, la capacità di percepire gli eventi storici nella loro dimensione locale, nazionale, europea e mondiale e di collocarli secondo le coordinate spazio-temporali, cogliendo nel passato le radici del presente. Se sul piano epistemologico i confini tra la storia, le scienze sociali e l'economia sono distinguibili, più frequenti sono le connessioni utili alla comprensione della complessità dei fenomeni analizzati. Comprendere la continuità e la discontinuità, il cambiamento e la diversità in una dimensione diacronica attraverso il confronto fra epoche e in dimensione sincronica attraverso il confronto fra aree geografiche e culturali è il primo grande obiettivo dello studio della storia. Il senso dell'appartenenza, alimentato dalla consapevolezza da parte dello studente di essere inserito in un sistema di regole fondato sulla tutela e sul riconoscimento dei diritti e dei doveri, concorre alla sua educazione alla convivenza e all'esercizio attivo della cittadinanza. La partecipazione responsabile - come persona e cittadino - alla vita sociale permette di ampliare i suoi orizzonti culturali nella difesa della identità personale e nella comprensione dei valori dell'inclusione e dell'integrazione. La raccomandazione del Parlamento e del Consiglio europeo 18 dicembre 2006 sollecita gli Stati membri a potenziare nei giovani lo spirito di intraprendenza e di imprenditorialità. Di conseguenza, per promuovere la progettualità individuale e valorizzare le attitudini per le scelte da compiere per la vita adulta, risulta importante fornire gli strumenti per la conoscenza del tessuto sociale ed economico del territorio, delle regole del mercato del lavoro, delle possibilità di mobilità.

(da Documento Tecnico allegato al DM. 22 agosto 2007 Regolamento recante norme in materia di adempimento dell'obbligo di istruzione)

3.5 Regolamento

Il Ministero dell'Istruzione, dell'Università e della Ricerca - Direzione Generale per gli Ordinamenti Scolastici e per l'Autonomia Scolastica - promuove, dall'anno scolastico 2008/2009, le Competizioni di Informatica denominate "Olimpiadi di problem solving", rivolte agli alunni della scuola dell'obbligo.

La competizione si propone di favorire lo sviluppo delle competenze generali di problem solving e valorizzare le eccellenze presenti nelle scuole, nonché sollecitare la diffusione dei contenuti scientifici e culturali dell'informatica come strumento di formazione nei processi educativi. Il problem solving rimanda a processi cognitivi in cui prevale il pensare, il ragionare, il fare ipotesi, attività che richiedono l'impiego di abilità relative alla gestione di informazioni strutturate più che l'applicazione di procedimenti meccanici volti all'esecuzione esclusiva di calcoli. Tali attività favoriscono l'acquisizione di competenze trasversali ai diversi contesti disciplinari riconosciute ormai essenziali per un inserimento attivo e consapevole dei giovani nella società. Le prove fanno riferimento anche a quelle adottate dalle indagini internazionali per rilevare gli apprendimenti e le capacità di problem solving, prove focalizzate sui processi piuttosto che sulle nozioni. Con queste competizioni si intende anche stimolare nei giovani l'interesse a sviluppare le capacità richieste in tutte le iniziative attivate per la valorizzazione delle eccellenze (Certamen, gare, competizioni, Olimpiadi di Matematica, Olimpiadi di Informatica).

Regolamento e modalità di partecipazione Le competizioni riguardano la scuola dell'obbligo e si svolgono su tre distinti livelli:

- Scuola primaria (alunni delle classi IV e V)
- Scuola secondaria di primo grado (alunni delle classi I, II e III)
- Scuola secondaria di secondo grado (alunni del primo biennio)

Le prove sono della stessa tipologia, ma il loro livello di difficoltà si differenzia a seconda del ciclo di istruzione.

Le competizioni si articolano in fase d'istituto, fase regionale e fase nazionale. Se i tempi interni dei singoli istituti lo permettono, queste gare sono precedu-

te da una fase di allenamento. L'unità di valutazione è la "squadra", ognuna costituita da quattro/cinque allievi. È vivamente raccomandata la rappresentanza di entrambi i sessi. Le prove hanno la durata di 90 minuti e consistono nella risoluzione di un insieme di problemi - comunemente circa una ventina - scelti dal Comitato organizzatore. La competizione è gestita da un sistema automatico sia per la distribuzione dei testi delle prove sia per la raccolta dei risultati e la loro correzione. L'adozione di questo sistema impone vincoli alla formulazione dei quesiti e delle relative risposte. Ulteriori informazioni sullo svolgimento delle diverse fasi delle competizioni vengono comunicate tempestivamente sulla pagina Internet di riferimento. Durante lo svolgimento delle prove (allenamenti, gare di istituto, gare regionali e finale) le squadre partecipanti possono servirsi anche di propri dispositivi digitali portatili collegati a Internet. Le scuole che intendono partecipare alle competizioni devono individuare un docente referente, il quale cura la loro registrazione. La pagina web di riferimento è <http://www.olimpiadiproblemsolving.it>.

3.6 Le prove

Il problem solving riguarda processi generali di ragionamento e di risoluzione di problemi ed è concepito non come una disciplina, ma come l'applicazione di conoscenze e abilità cui si ricorre nell'affrontare situazioni problematiche in contesti reali. In particolare, riguarda la comprensione della natura del problema, l'identificazione degli elementi che lo compongono e le loro interrelazioni, la scelta di rappresentazioni adeguate nella risoluzione di situazioni problematiche, la riflessione sulla soluzione individuata e la sua efficace comunicazione. Le prove somministrate durante gli allenamenti e le successive competizioni richiedono l'impiego e lo sviluppo delle competenze fondamentali tipiche del problem solving:

- Ricerca: definizione dei dati del problema, formulazione dell'ipotesi e individuazione delle fonti per il reperimento di ulteriori dati e informazioni;
- Strutturazione: rappresentazione del contesto attraverso dati, tabelle e grafici, etc.

- Esplorazione: studio di casi particolari in cui la soluzione è semplice o particolarmente significativa;
- Analisi: scomposizione in sottoproblemi e scoperta del processo di risoluzione da utilizzare;
- Elaborazione: algoritmo di risoluzione del problema, progettazione e programmazione;
- Verifica: controllo dei risultati ottenuti e scelta della strategia più efficace;
- Comunicazione: presentazione e condivisione delle informazioni relative alle tematiche affrontate e alla soluzione

Per consentire la conoscenza dei contenuti e l'approccio metodologico della competizione, sono predisposti esempi di prove cui possono accedere tutte le squadre che le scuole partecipanti intendono iscrivere. Gli argomenti su cui vertono le prove sono riferiti alle diverse aree disciplinari e agli assi culturali. Le prove constano di circa 20 esercizi che spaziano dall'italiano all'inglese, e dalla matematica a quesiti su alberi binari di ricerca, grafi e storia. Le categorie nelle quali sono stati raggruppati gli esercizi sono le seguenti:

Italiano	Grafi
Storia	Tabelle
Inglese	Crittografia
Matematica	Torre di Hanoi
Combinatoria	Knapsack
Statistica	Pianificazione (o programmazione)
Deduzione	Project management

In seguito verranno presentati alcuni fra gli esercizi più significativi proposti durante le gare. Per ogni singola risoluzione sarà necessario acquisire una particolare, nonché (apparentemente) specifica, competenza. Con un'attenta analisi

sarà possibile notare una somiglianza abbastanza forte fra le abilità cognitive strettamente legate al campo di pertinenza dell'esercizio, e lo sviluppo di concetti e modi operandi che nell'informatica sono usuali e fondamentali. Inoltre, in aggiunta a ciò, i ragazzi si troveranno all'interno di un contesto che stimola l'attività di gruppo, una mentalità orientata alla divisione dei compiti e al rispetto delle scadenze, e l'aiuto reciproco al fine di raggiungere un obiettivo comune.

Esercizio di Torre di Hanoi. Si ha una tavoletta con tre pioli numerati come in Figura 3.1. Sul piolo 3 c'è una pila di dischi, su ciascuno dei quali è inciso il nome di una città italiana. Si possono spostare i dischi solamente uno alla volta e ogni spostamento costituisce una mossa. Qual è il numero minimo di mosse M necessarie per trasferire i dischi sul piolo 1 in modo che dall'alto in basso la longitudine delle città sia crescente?

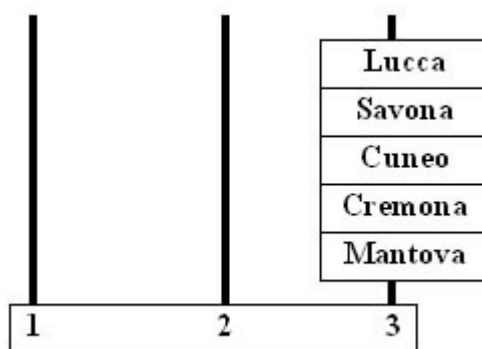


Figura 3.1: Torre di Hanoi

Elementi cognitivi. Un gioco, all'apparenza semplice e quasi infantile, come la Torre di Hanoi, nasconde in realtà, per la sua risoluzione, un procedimento matematico di grande interesse: la ricorsione. La Torre di Hanoi è composta da una serie di dischi, di dimensioni decrescenti, in numero variabile. Il gioco consiste nel demolire la torre e nel ricostruirla su un'altra colonnina, seguendo delle regole ben precise. Con un po' di pratica si arriva facilmente a capire il procedimento da seguire con un numero qualsiasi di dischi, scoprendo la formula risolutiva del gioco. La Tabella 3.4 mostra il numero di mosse minime di mosse necessarie per

la risoluzione. È utile mettere in evidenza i valori in base due della tabella che costituiscono una successione di grande semplicità: ogni volta che si aggiunge un disco alla torre, basta aggiungere un 1 al numero dei movimenti necessari.¹ A titolo informativo si riporta un algoritmo ricorsivo per la risoluzione della torre:

Caso base ($n = 1$). Spostare il disco dalla sorgente Src alla destinazione Dst

Caso ricorsivo. Spostare $n - 1$ dischi da Src a un piolo intermedio $Src1$, spostare il rimanente disco in Dst , spostare gli $n - 1$ dischi da $Src1$ a Dst .

Tabella 3.4: Tabella raffigurante il numero minimo di mosse necessarie per la risoluzione di una Torre di Hanoi con n dischi.

Dischi	1	2	3	4	5	6	7	n
$Mosse_{10}$	1	3	7	15	31	63	127	$2^n - 1$
$Mosse_2$	1	11	111	1111	11111	111111	1111111	...

Esercizio di Project management. Alcuni ragazzi decidono di costruire un file multimediale sugli avvenimenti storici significativi della loro regione. Per organizzare il progetto, dividono il lavoro in 9 attività e assegnano ognuna di queste a un gruppo di loro. La Figura 3.2 descrive le 9 attività (indicate rispettivamente con le sigle A1, A2,...,A9), riportando per ciascuna di esse il numero di ragazzi assegnato e il numero di giorni necessario per completarla. Le priorità fra le attività sono descritte con coppie di sigle; ogni coppia esprime il fatto che l'attività associata alla sigla di destra possa iniziare solo quando l'attività associata alla sigla di sinistra sarà terminata. L'attività che non ha priorità è la prima, quella che non ha successori è l'ultima. Questo è l'elenco delle coppie: (A1,A2), (A1,A3), (A3,A4), (A3,A5), (A3,A6), (A2,A7), (A5,A7), (A4,A8), (A5,A8), (A6,A9), (A7,A9), (A8,A9). Trovare quanti giorni N sono necessari per completare il progetto, tenuto presente che alcune attività possono essere svolte in parallelo e che ogni attività deve iniziare il prima possibile (nel rispetto delle

¹Rimanendo necessario provare la validità della soluzione data, ed è possibile provarla applicando il principio di induzione matematica.

priorità date). L'attività A1 inizia il giorno 1; trovare il numero X1 che individua il giorno in cui lavora il maggior numero M1 di ragazzi, e il numero X2 del giorno in cui lavora il minor numero M2 di ragazzi. Supponendo che la retribuzione media giornaliera per ragazzo sia di 90 euro, calcolare il costo complessivo S del progetto.

attività	ragazzi	giorni
A1	5	1
A2	2	2
A3	2	1
A4	2	2
A5	3	1
A6	4	3
A7	3	2
A8	5	1
A9	5	1

Figura 3.2: Tabella delle attività

Elementi cognitivi. Con questi esercizi si vogliono simulare gestione e conduzione di un progetto di gruppo che comprende diverse attività. Aumentando leggermente le dimensioni del problema crescerà molto rapidamente il numero di sotto-attività da portare a termine, rendendo indispensabile la collaborazione di gruppo. Gestione del tempo, suddivisione problema in sottoproblemi, suddivisione compiti, rispetto delle deadline, raccolta e assemblamento dei risultati sono le abilità con cui è necessario prendere dimestichezza.

Esercizio sul problema dello zaino. Nelle lezioni di educazione alimentare i ragazzi hanno classificato alcuni alimenti in relazione al contenuto proteico e al loro costo. I risultati di questa classificazione sono stati descritti da una tabella avente la dichiarazione

$$\text{tabx}(\text{sigla dell'alimento}, \text{contenuto proteico}, \text{costo in euro})$$

Il contenuto della tabella riporta i dati relativi a 7 alimenti ed è il seguente:

$$\text{tabx}(m1, 30, 53) \quad \text{tabx}(m2, 16, 27) \quad \text{tabx}(m3, 24, 42)$$

$$tabx(m4, 37, 49) \quad tabx(m5, 43, 78) \quad tabx(m6, 68, 81) \quad tabx(m7, 72, 99)$$

Trovare le risposte ai seguenti quesiti; se la risposta è una lista, riportare le sigle degli alimenti in ordine crescente; per le sigle si ha il seguente ordine $m1 < m2 < m3 < \dots < m7$. Con gli elementi sopra descritti trovare la lista L degli elementi utilizzati per costruire una dieta con valore proteico di almeno 180 e un costo non superiore a 250 euro; trovare inoltre il valore proteico P di questa dieta e il suo costo Q.

Elementi cognitivi. Il problema dello zaino, detto comunemente knapsack problem, è un problema di ottimizzazione combinatoria la cui soluzione generale non è possibile da raggiungere in tempo polinomiale in tutti i casi. Esso, infatti, è NP-hard, e fa parte dell'insieme dei problemi NP-completi. Le istanze del problema dello zaino proposte durante le gare sono tuttavia risolvibili mediante l'applicazione di un ragionamento ricorsivo più semplice.

Esercizio sui grafi. Un grafo stradale è costituito da città e strade che le congiungono; il termine

$$a(citta1, citta2, distanza)$$

descrive un tratto stradale che unisce la *citta1* e *citta2* con l'indicazione della relativa distanza (per esempio in chilometri). Un percorso tra due città di questo grafo viene descritto con la lista delle città attraversate, ordinate da quella di partenza a quella di arrivo; la sua lunghezza è la somma delle distanze dei tratti che uniscono due città successive. Sia dato il grafo stradale descritto dai seguenti tratti:

$$\begin{aligned} a(n5, n1, 2) \quad a(n5, n3, 5) \quad a(n3, n8, 4) \quad a(n2, n8, 6) \\ a(n4, n6, 1) \quad a(n6, n7, 5) \quad a(n1, n7, 3) \quad a(n3, n7, 9) \\ a(n2, n6, 4) \quad a(n7, n5, 7) \quad a(n4, n7, 7) \quad a(n1, n4, 6) \end{aligned}$$

Disegnare il grafo (in modo che gli archi non si incrocino) e trovare:

1. La lista L1 del percorso più lungo (senza passare più di una volta per uno stessa città) dalla città n3 alla n1 e calcolare la sua lunghezza K1

2. La lista L2 del percorso dalla città n3 alla n1 che passa una sola volta per tutte le città del grafo e calcolarne la lunghezza K2

Elementi cognitivi. I grafi sono una struttura matematica molto usata nelle applicazioni e si presentano a rappresentare problemi apparentemente molto diversi tra loro con un linguaggio semplice, non ambiguo ed unificato. Chi si appresta per la prima volta alla risoluzione di questo tipo di problemi si allenerà al ragionamento logico e imparerà la visita di un grafo, il riconoscimento dei cicli, delle semplici operazioni di calcolo di percorsi.

Esercizio di matematica. Sul fianco di una montagna esistono numerose sorgenti. L'acqua di una sorgente, che si suppone fluire in modo costante, può scorrere a valle attraverso uno o più rigagnoli. Può avvenire che questi convergano in un punto in cui esiste una sorgente; in tal caso, la loro acqua si aggiunge a quella fornita da questa sorgente. La situazione è quindi descrivibile con una rete: i nodi della rete rappresentano le sorgenti e gli archi rappresentano i rigagnoli. La situazione complessiva è quindi reticolo:

- $s(\text{sorgente}, \text{litri d'acqua al minuto})$
specifica l'acqua che sgorga da ogni sorgente
- $r(\text{sorgente1}, \text{sorgente2})$
specifica il rigagnolo dalla sorgente n.1 alla sorgente n.2

Se da una sorgente escono più rigagnoli, l'acqua si divide in parti uguali fra ciascuno di essi. Si veda l'esempio in Figura 3.3 dove

$$\begin{array}{cccccc} s(a, 6) & s(b, 5) & s(c, 1) & s(d, 4) & s(e, 3) & s(f, 2) \\ r(a, c) & r(a, d) & r(b, d) & r(c, e) & r(d, e) & r(d, f) \end{array}$$

La quantità d'acqua che sgorga dai nodi c , e e f è 4, 13 e 8 rispettivamente.

Elementi cognitivi. Questi semplici esercizi consentono una maggiore confidenza con il ragionamento logico-matematico. La semplicità delle operazioni richieste per la risoluzione permettono allo studente di dedicarsi con cura ad una prima, essenziale rappresentazione grafica del contesto, utile anche come punto di ritrovo per successivi errori di comprensione.

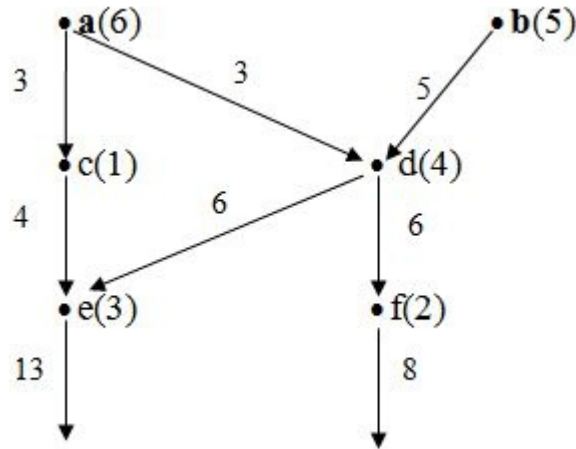


Figura 3.3: Esercizio di matematica: sorgenti e rigagnoli

Esercizio di matematica. Si devono consegnare delle pizze alle abitazioni poste ai numeri dispari di una strada. Per rispettare i tempi delle prenotazioni, le pizze devono essere consegnate seguendo le istruzioni scritte usando un codice che specifica come spostarsi avanti e indietro (per esempio (a, x) e (i, x) , con x che specifica il numero di passi) lungo la strada a partire da un punto specificato. Un esempio di consegna di 4 pizze: se a partire dalla casa al numero 1 le istruzioni fossero descritte dalla lista $(a, 2)$, $(a, 1)$, $(i, 2)$, le consegne seguirebbero l'ordine dei numeri civici 1, 5, 7, 3. A partire dalla casa al numero 3, con le istruzioni $(a, 1)$, $(i, 2)$, $(a, 4)$ le consegne seguirebbero l'ordine 3, 5, 1, 9.

1. Si devono consegnare 8 pizze ad alcune abitazioni che corrispondono ai numeri civici dispari di una strada. Le istruzioni per la consegna, a partire dalla abitazione al numero 5, sono le seguenti: $(a, 4)$, $(a, 4)$, $(i, 5)$, $(a, 2)$, $(i, 6)$, $(a, 3)$, $(a, 4)$. Trovare la lista L1 che contiene i numeri civici delle abitazioni disposti secondo l'ordine di consegna delle pizze.
2. Si devono consegnare 8 pizze ad alcune abitazioni che corrispondono ai numeri civici dispari di una strada. Le istruzioni per la consegna, a partire dalla abitazione al numero 7, sono le seguenti: $(a, 7)$, $(i, 3)$, $(i, 6)$, $(a, 4)$, $(i, 3)$, $(a, 2)$, $(a, 4)$. Trovare la lista L2 che contiene i numeri civici delle abitazioni disposti secondo l'ordine di consegna delle pizze.

Elementi cognitivi. È richiesto di operare su semplici liste di elementi.

Esercizio di matematica. Data una lista di numeri diversi, per esempio $[1, 5, 2, 4]$, è possibile alterare l'ordine dei suoi elementi scambiando di posto due elementi adiacenti. Con mosse successive è dunque possibile spostare gli elementi della lista in modo da ottenere i numeri disposti in ordine crescente; in questo esempio, l'ordinamento si ottiene con due mosse (gli elementi adiacenti da scambiare sono in grassetto)

Prima mossa da $[1, 5, 2, 4]$ a $[1, 2, 5, 4]$

Seconda mossa: da $[1, 2, 5, 4]$ a $[1, 2, 4, 5]$

La lista $[2, 5, 4, 3]$ può essere ordinata con tre mosse:

$$[2, 5, 4, 3] \rightarrow [2, 4, 5, 3] \rightarrow [2, 4, 3, 5] \rightarrow [2, 3, 4, 5]$$

Date le liste $L1 = [9, 8, 7, 6, 1, 5, 4, 3, 2]$, $L2 = [9, 8, 2, 1, 7, 6, 4, 3, 5]$ ed $L3 = [5, 7, 4, 6, 3, 9, 2, 8, 1]$, trovare il numero minimo di mosse ($N1$, $N2$, $N3$ rispettivamente per $L1$, $L2$, $L3$) necessario per ottenere le corrispondenti permutazioni ordinate crescenti.

Elementi cognitivi. In questo esercizio è richiesta l'individuazione di una semplice tecnica di ordinamento di una lista. Questi concetti costituiscono la base degli algoritmi di ordinamento più complessi, nel caso specifico i cosiddetti algoritmi di ordinamento *con scambio* come Quicksort, Bubble sort, Cocktail sort, Odd-even sort, Comb sort, etc.

Esercizio di programmazione. In un foglio a quadretti disegnare un rettangolo di 14 quadretti in orizzontale e 9 in verticale (vedi Figura 3.4). Ogni casella può essere individuata da due numeri interi; per esempio la casella contenente 'P' è individuata dall'essere nella sesta colonna (da sinistra) e nella terza riga (dal basso): brevemente si dice che ha coordinate $(6, 3)$. Le coordinate della casella contenente 'S' sono $(10, 8)$ e di quella contenente la freccia sono $(1, 1)$. La freccia è una piccola tartaruga, in questo caso voltata verso destra, che può eseguire tre tipi di comandi:

- Girarsi di 90 gradi in senso orario: comando o
- Girarsi di 90 gradi in senso antiorario: comando a
- Avanzare di una casella (nel senso della freccia): comando f

Questi comandi possono essere concatenati in sequenza in modo da permettere alla tartaruga di compiere vari percorsi; per esempio, la lista f, f, f, f, f, a, f, f fa spostare la tartaruga dalla posizione e orientamento iniziali mostrati in Figura 3.4 fino alla casella “P”; risultato analogo si ottiene con la lista $a, f, f, o, f, f, f, f, f$. Tuttavia, nel primo caso l’orientamento finale della tartaruga è verso l’alto, mentre nel secondo caso l’orientamento finale è verso destra.

In un rettangolo 14×9 la tartaruga è nella casella $(6, 4)$ ed è orientata verso destra. Trovare l’ascissa X e l’ordinata Y in cui si troverà la tartaruga dopo aver effettuato il percorso descritto dalla lista $f, o, f, a, f, a, f, f, a, f, f, a, f, o, f, o, f, f, a, f, o, f, a, f, f, o, f, f, o, f, o, f, f$ e l’orientamento D descritto con n, s, e, o rispettivamente per alto (nord), basso (sud), destra (est), sinistra (ovest).

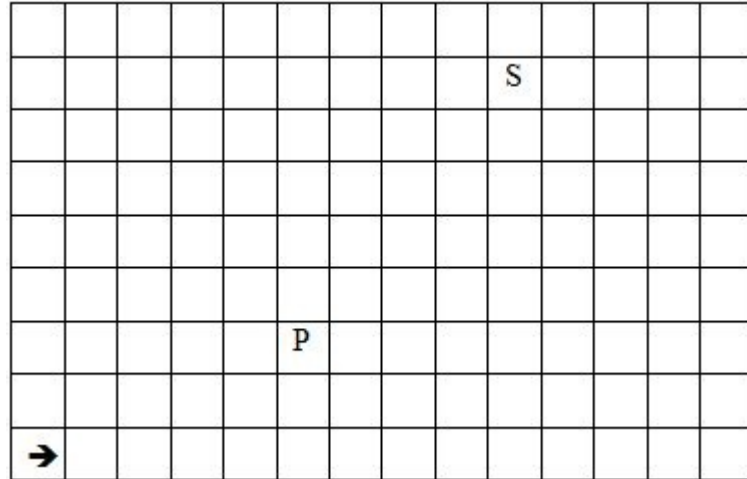


Figura 3.4: Piano della tartaruga

Elementi cognitivi. L’obiettivo di questi esercizi elementari è quello di far conoscere agli studenti le attività legate alla programmazione, ovvero l’esecuzione iterativa di un insieme di azioni, il calcolo di soluzioni alternative più o meno

ottimizzate, il controllo delle condizioni e la verifica dei casi limite. Tutte pratiche che si rendono assolutamente utili a tutti, non soltanto a quelle minoranze che intraprenderanno la carriera informatica.

Esercizio di deduzione. Si ricorda che il termine

$$\text{regola}(\text{sigla}, \text{lista_precedenti}, \text{successivo})$$

descrive una regola che consente di dedurre l'elemento successivo conoscendo tutti quelli contenuti nella lista dei precedenti; ogni regola è poi identificata in modo univoco da una sigla. Dato l'insieme di regole

$$\begin{aligned} & \text{regola}(1, [c1, c2], ip) \quad \text{regola}(2, [i, h], a) \quad \text{regola}(3, [h, p1], c1) \\ & \text{regola}(4, [h, p2], c2) \quad \text{regola}(5, [c1, c2], a) \quad \text{regola}(6, [p1, p2], h) \\ & \text{regola}(7, [p1, p2], i) \quad \text{regola}(8, [c1, i], c2) \quad \text{regola}(9, [i, a], h) \\ & \text{regola}(10, [h, c1], p1) \quad \text{regola}(11, [h, c2], p2) \quad \text{regola}(12, [c1, a], c2) \\ & \text{regola}(13, [p1, h], p2) \quad \text{regola}(14, [p1, i], p2) \quad \text{regola}(15, [c2, i], c1) \\ & \text{regola}(16, [a, h], i) \quad \text{regola}(17, [p1, c1], h) \quad \text{regola}(18, [p2, c2], h) \\ & \text{regola}(19, [c2, a], c1) \quad \text{regola}(20, [p2, h], p1) \quad \text{regola}(21, [p2, i], p1) \end{aligned}$$

si osserva, conoscendo gli elementi contenuti nella lista $[p1, p2]$, che è possibile ad esempio dedurre o calcolare direttamente h con la regola 6 e i con la regola 7; ma conoscendo $[p1, p2]$ è anche possibile dedurre $c1$ applicando prima la regola 6 (per dedurre h) e poi la regola 3 (conoscendo ora $[h, p1]$). Si può quindi dire che la lista $[6, 3]$ rappresenta un procedimento per dedurre o calcolare $c1$ da $[p1, p2]$; la lista contiene infatti l'indicazione delle regole che devono essere applicate. Per esempio, la lista $[6, 3, 4, 5]$ rappresenta il procedimento per calcolare a da $[p1, p2]$.

Coi dati $[c1, h]$ esistono due procedimenti descritti da due liste L1 e L2 per calcolare i . Il primo procedimento è descritto da L1 = $[10, 13, 7]$; trovare la lista L2 che descrive il secondo.

Esercizio di Crittografia. Le stringhe sotto riportate contengono, crittografate col metodo Giulio Cesare, il nome di un noto e importante personaggio della storia romana e il riferimento a un luogo cui egli è legato; queste due informazioni sono crittografate con due chiavi diverse

Nome: [f,h,v,d,u,h]

Luogo: [q,t,a,h,b,n,m,d]

Trovare le chiavi N e M con cui sono state crittografate le due informazioni sul personaggio, sapendo che l'alfabeto utilizzato è: a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z. Si ricorda di non considerare eventuali caratteri non alfabetici (virgola, punto, apostrofo, etc.)

Capitolo 4

Analisi di dati sperimentali

La sperimentazione cui si fa riferimento (Olimpiadi di Problem Solving) è iniziata a partire dall'anno scolastico 2009/2010. Ad oggi si contano 672 istituzioni scolastiche suddivise equamente fra scuole primarie, scuole secondarie di primo e secondo grado (di quest'ultima si considera soltanto il primo biennio). Tenendo conto del fatto che, secondo il regolamento, ogni istituzione scolastica può iscrivere un numero di squadre non inferiore a tre e non superiore a otto, il database ospiterà i dati relativi a un numero di squadre che va da 2000 a - potenzialmente - 5300 squadre. Si sta trattando dunque una mole di dati tutto sommato significativa, che raccoglie il comportamento di 10000/25000 ragazzi. Il progetto è stato accolto con entusiasmo in Veneto, Sicilia, Puglia, Lombardia ed Emilia Romagna, dove si contano fino a 100 scuole partecipanti, uniformemente distribuite per ciclo di istruzione, per ogni regione. Poco effetto ha invece suscitato in Friuli Venezia Giulia, Sardegna, Trentino Alto Adige, Umbria e Valle d'Aosta.

La fase di raccolta e interpretazione dei risultati delle gare effettuate dagli studenti che hanno partecipato alle olimpiadi ha riservato non poche sorprese, facendo comprendere che la valutazione del fenomeno "Olimpiadi di problem solving" debba tassativamente prescindere da un'accurata scelta, nonché tipizzazione dei problemi da sottoporre - prima - e un preciso sistema di raccolta e catalogazione dei compiti - poi. In seguito saranno prese in rassegna le mancanze della raccolta/catalogazione attuale.

Verranno ora presentati due tipi di valutazioni: la prima riguardante tutti i partecipanti, la seconda riservata alle cosiddette "eccellenze", cioè alle squa-

dre che sono risultate vincitrici delle selezioni regionali. Saranno discusse le problematiche riscontrate.

4.1 Valutazioni per categorie di problemi

Ai ragazzi sono stati sottoposti esercizi di varia natura e difficoltà, simili a quelli d'esempio presentati nel capitolo precedente. Le elaborazioni eseguite riportano l'andamento delle valutazioni medie ottenute da tutte le squadre nelle prove svolte in ciascun anno scolastico. Per la scuola primaria e per quella secondaria di primo grado sono disponibili i dati di due anni scolastici; per la secondaria di secondo grado sono disponibili i dati di un solo anno scolastico. Quindi, in sintesi, saranno disponibili cinque analisi temporali per ogni scelta di discipline oggetto della verifica. Le scelte fatte sono le seguenti:

- i) Problemi di comprensione linguistica (italiano e inglese)
- ii) Problemi tipo knapsack¹ e percorsi su grafi (oltre a rappresentare una novità in fatto di esercizi, introducono una serie di concetti e abilità cognitive di base fra le più significative nonchè ricorrenti in qualsiasi campo dell'informatica)
- iii) Problemi di cultura generale
- iv) Problemi di project management
- v) Tutti i problemi complessivamente

Scuola primaria

La Figura 4.1 mostra l'andamento generale di tutte le scuole primarie che hanno partecipato al progetto durante gli anni scolastici 2009/2010 e 2010/2011.

¹Il problema dello zaino, detto comunemente *knapsack problem*, è un problema di ottimizzazione combinatoria la cui soluzione generale non è possibile da raggiungere in tempo polinomiale in tutti i casi. Esso, infatti, è NP-hard, e fa parte dell'insieme dei problemi NP-completi. Le istanze del problema dello zaino proposte durante le gare sono tuttavia risolvibili mediante l'applicazione di un ragionamento ricorsivo più semplice.

A parte qualche lieve oscillazione, il trend tende al positivo in entrambi gli anni. Le Figure 4.2 e 4.3 mostrano invece l'andamento, specifico e aggregato, per i soli esercizi di italiano e inglese. Come si può notare in questi ultimi due casi l'ultima prova ha registrato percentuali di correttezza inferiori alla media, invertendo il trend positivo che ci si poteva aspettare. In particolare i quesiti dell'ultima prova del 2011 hanno registrato un livello di difficoltà anomalo, spiegando dunque il motivo della discesa peraltro inaspettata in quanto trattasi della prova finale. La Figura 4.4 mostra l'andamento rispetto ai quesiti di matematica. Se per l'anno 2010/2011 è possibile constatare un se pur moderato miglioramento, ben poco si comprende dal grafico relativo al 2010. Ciò si spiegherebbe con la mancanza di una competizione della fase di qualificazione unita ad una scelta poco equilibrata dei quesiti, inconvenienti che fanno parte di una sperimentazione. La Figura 4.5 pare abbia registrato il miglioramento più significativo fra tutte le categorie di problemi. Partendo infatti da uno stato iniziale con una bassissima percentuale di successi, che trova spiegazione nella novità rappresentata dagli esercizi di stampo puramente informatico come le operazioni sui grafi e il ragionamento ricorsivo, i ragazzi hanno ottenuto, nel corso dell'intera olimpiade, un discreto miglioramento. Per ultima, Figura 4.6, è raffigurata la valutazione rispetto ai quesiti di tutta l'area scientifica.

Scuola secondaria di primo grado

La Figura 4.7, come in precedenza, mostra i risultati ottenuti dalle indagini svolte nelle scuole secondarie di primo grado. Sono state selezionate le Figure 4.8, 4.9 per le valutazioni sui quesiti umanistico-linguistici, e le Figure 4.11, 4.10 - di particolare rilevanza considerata la novità rappresentata della categoria e il risultato ottenuto - e 4.12 per quelle appartenenti all'area scientifica. I dati sulle categorie non riportate si sono dimostrati poco significativi non tanto per la mancata manifestazione del miglioramento, quanto per l'andamento altalenante dovuto principalmente a due fattori: la mancanza di questo tipo di esercizi in alcune prove, l'eccessiva facilità/difficoltà non facilmente delineabile.

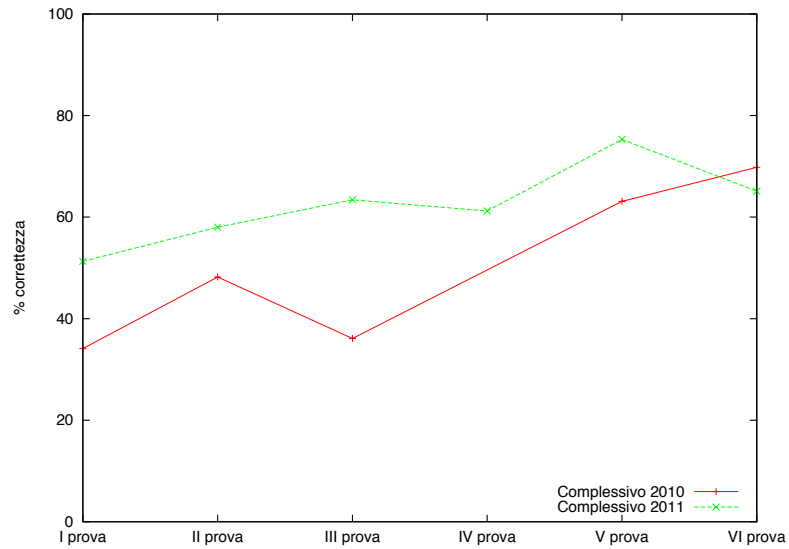


Figura 4.1: Situazione generale della scuola primaria

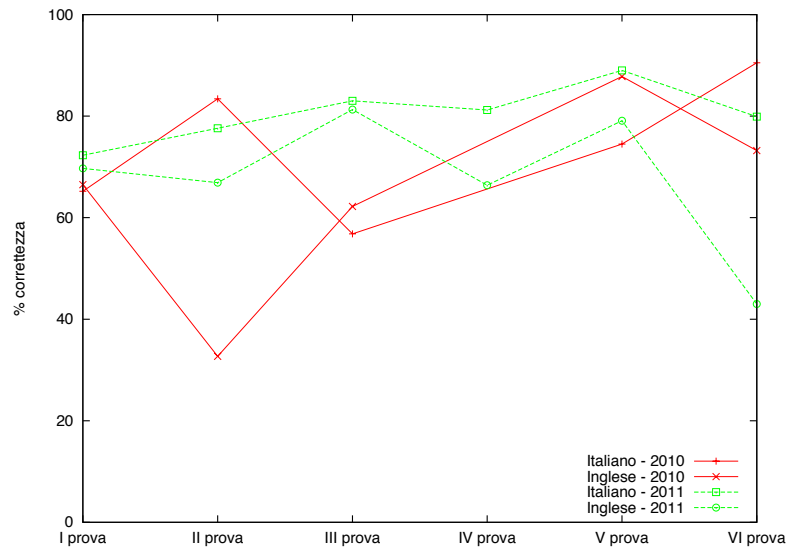


Figura 4.2: Italiano e inglese nelle scuole primarie

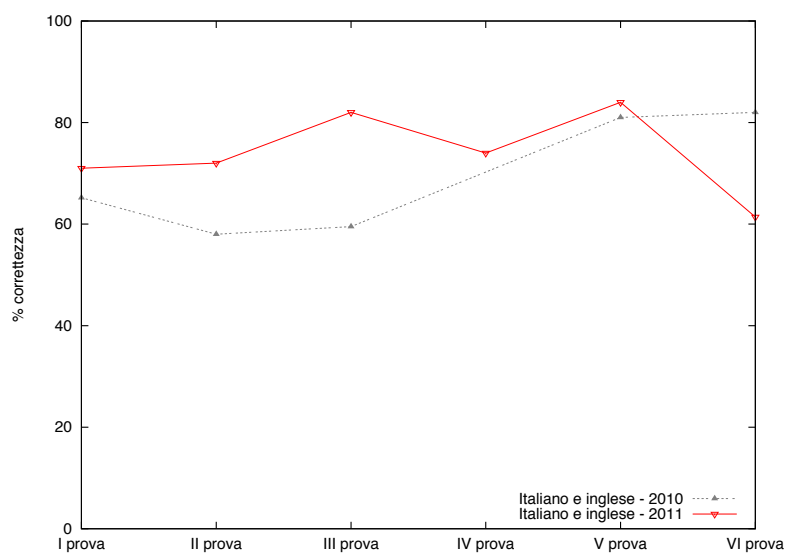


Figura 4.3: Aggregazione di *italiano* e *inglese* nelle scuole primarie

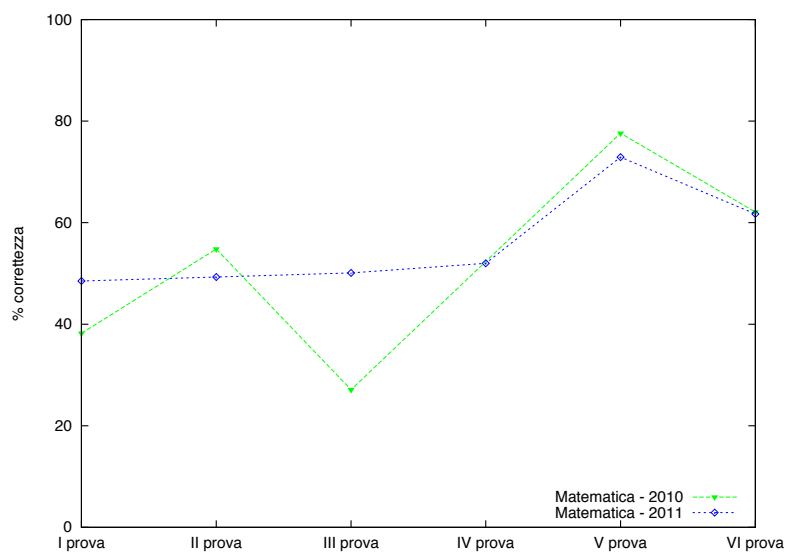


Figura 4.4: *Matematica* nelle scuole primarie

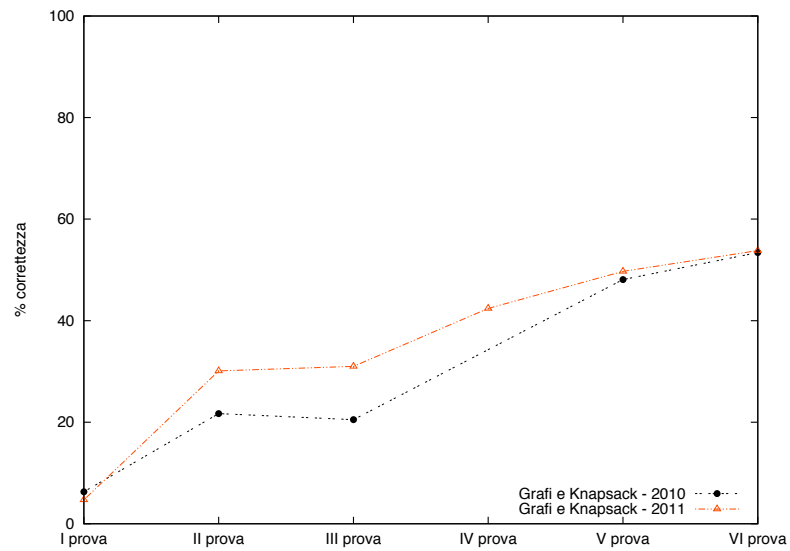


Figura 4.5: *Grafici e knapsack* nelle scuole primarie

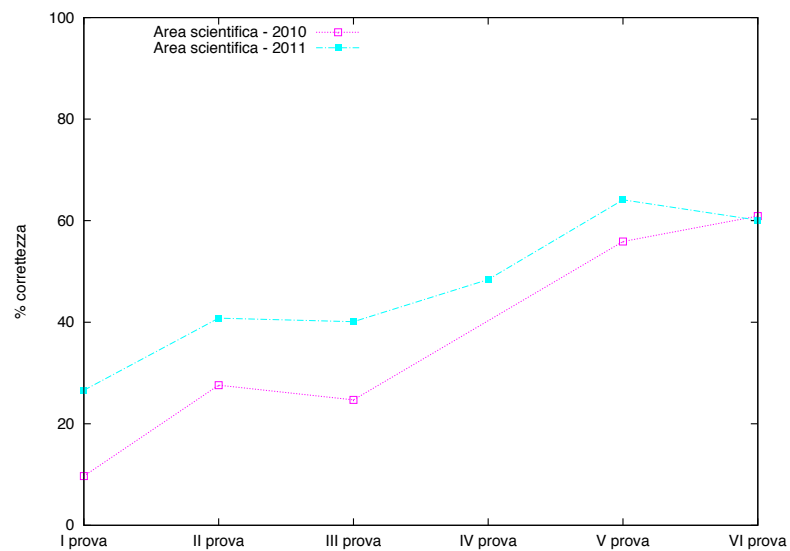


Figura 4.6: L'area scientifica nelle scuole primarie

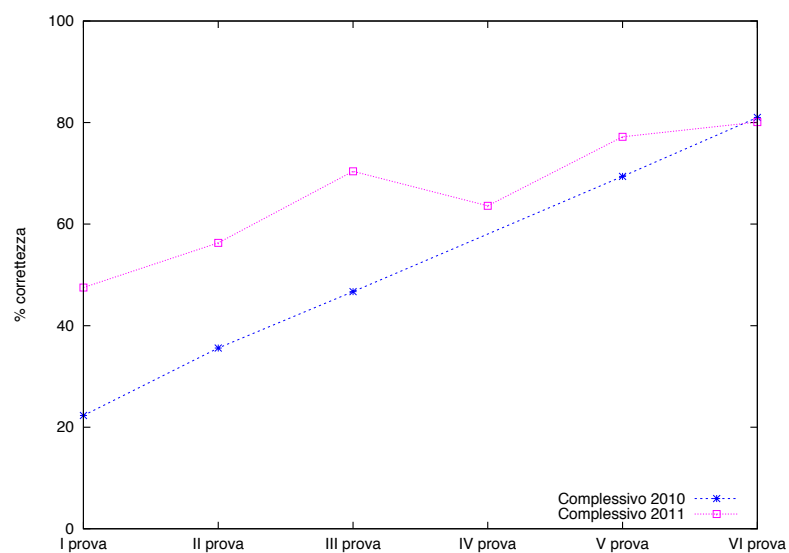


Figura 4.7: Situazione generale della scuola secondaria di primo grado

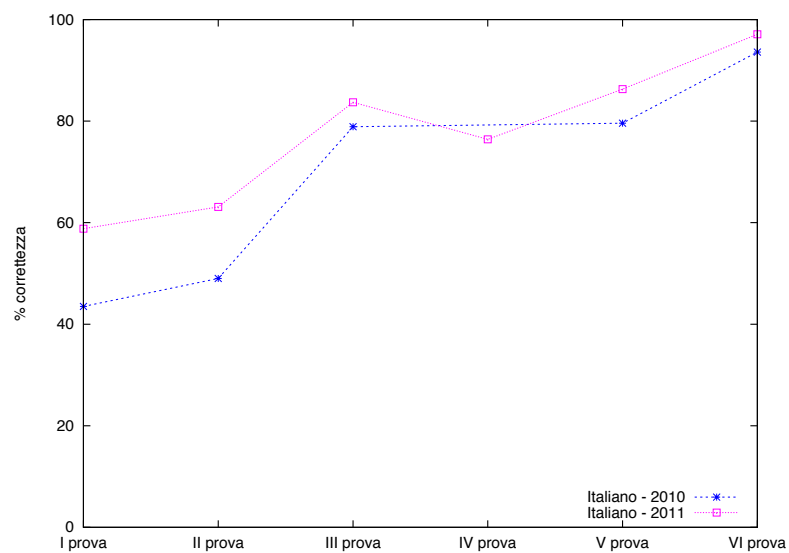


Figura 4.8: Italiano nella scuola secondaria di primo grado

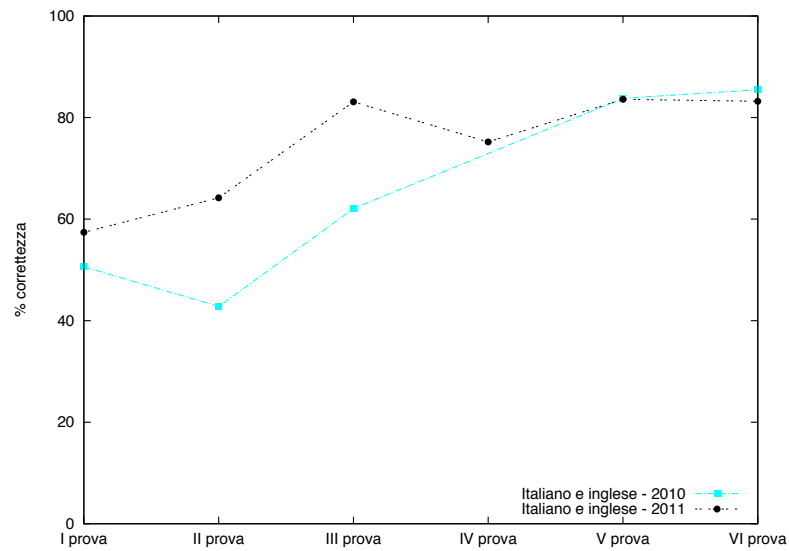


Figura 4.9: Aggregazione di *Italiano e inglese* nella scuola secondaria di primo grado

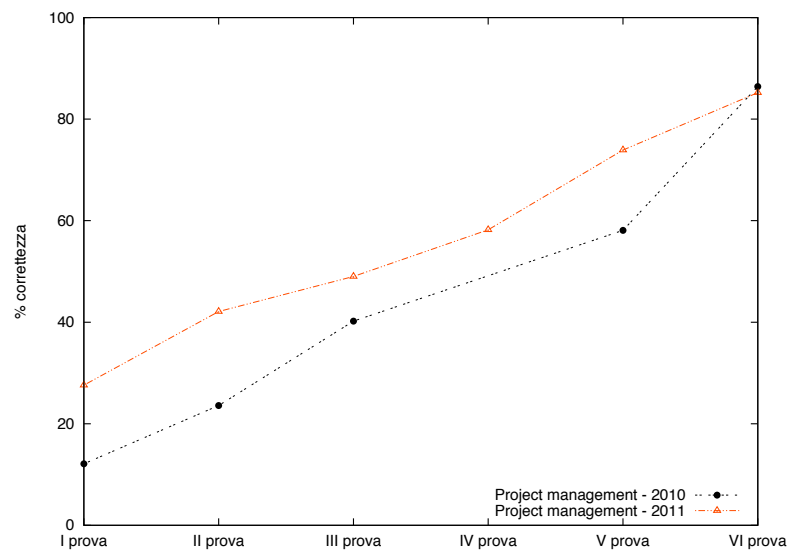


Figura 4.10: *Project management* nella scuola secondaria di primo grado

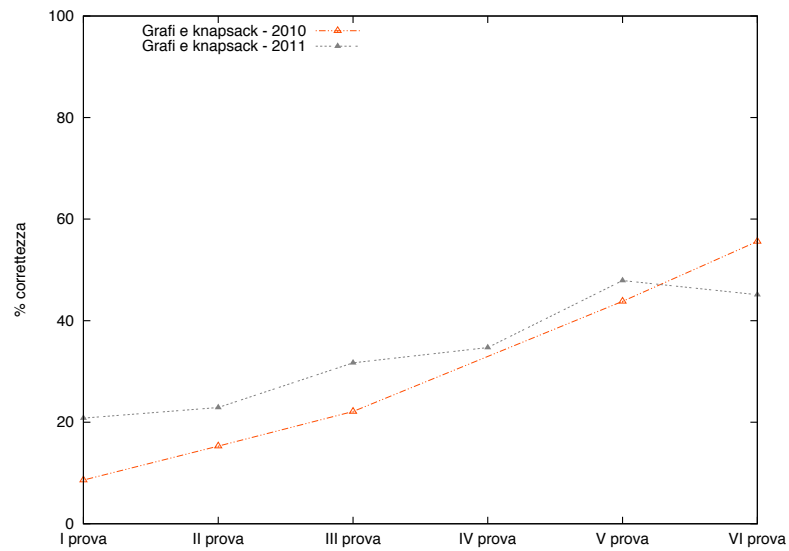


Figura 4.11: *Grafici e knapsack* nella scuola secondaria di primo grado

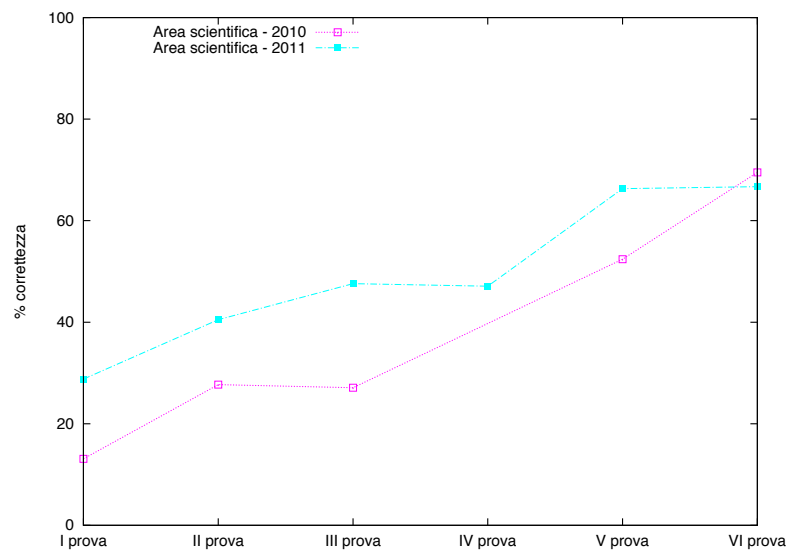


Figura 4.12: L'area scientifica nella scuola secondaria di primo grado

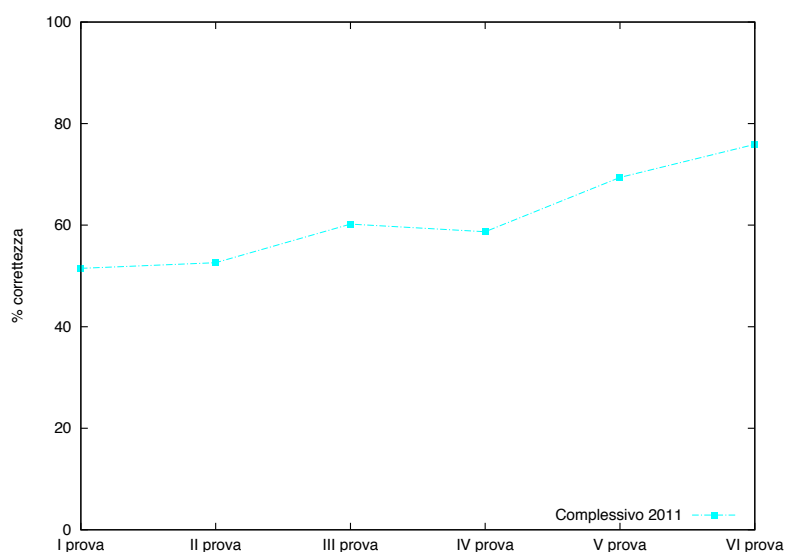


Figura 4.13: Situazione generale della scuola secondaria di secondo grado

Scuola secondaria secondo grado

Nei due anni obbligatori di scuola secondaria di secondo grado la sperimentazione è iniziata l'anno scolastico 2010/2011, motivo per cui non è stato possibile effettuare confronti con gli anni precedenti. Ci si è limitati dunque a considerare l'andamento complessivo durante tale anno, Figura 4.13, distinguendo l'area linguistica, Figura 4.14, da quella scientifica, Figura 4.15, e un confronto fra le due, Figura 4.16. I grafici di project management, grafi, matematica e pianificazione mostrano un miglioramento costante nel tempo, uno dei migliori risultati che si potessero ottenere. Si registra, al contrario e soltanto in questo ciclo di istruzione superiore, un peggioramento che riguarda le categorie knapsack e, con meno evidenza, deduzione. Siccome è da escludere una diminuzione delle capacità cognitive degli studenti, si è portati ad attribuire il motivo di questo comportamento alla difficoltà eccessivamente crescente degli esercizi oppure, da non escludere poichè fenomeno preoccupante, un calo di interesse, motivazione, curiosità.

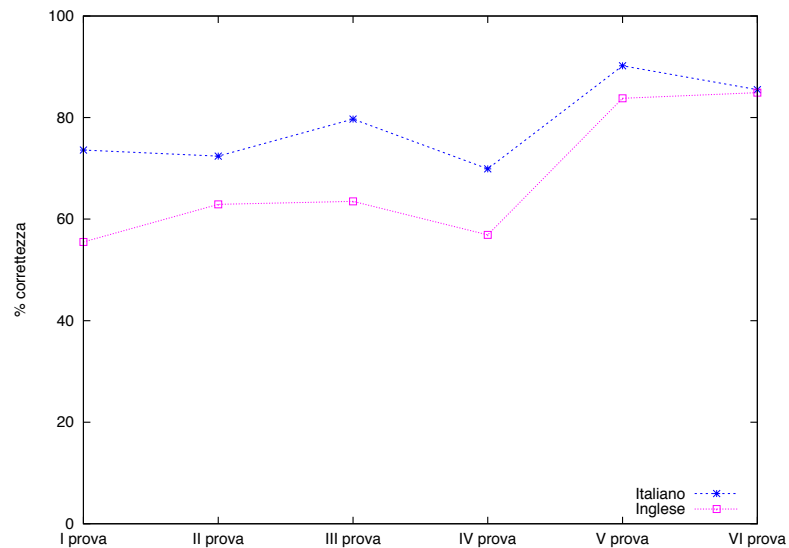


Figura 4.14: Italiano e inglese nelle scuole secondarie di secondo grado

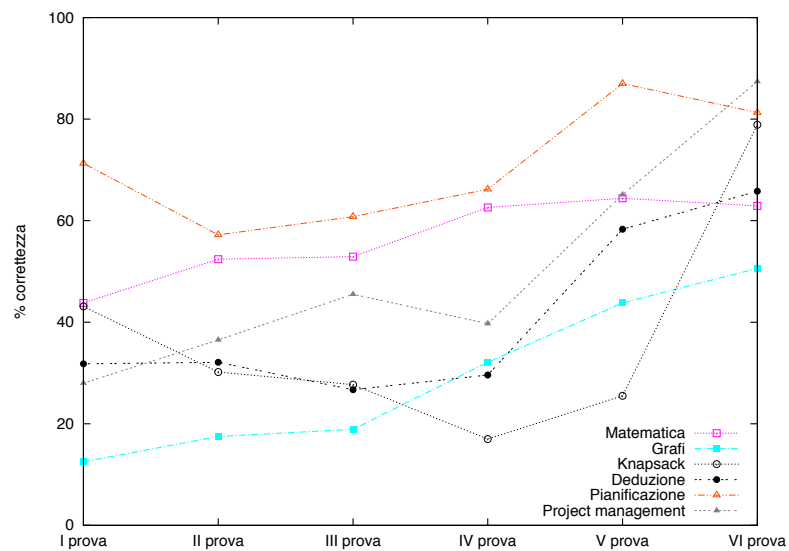


Figura 4.15: Categorie dell'area scientifica nelle scuole secondarie di secondo grado

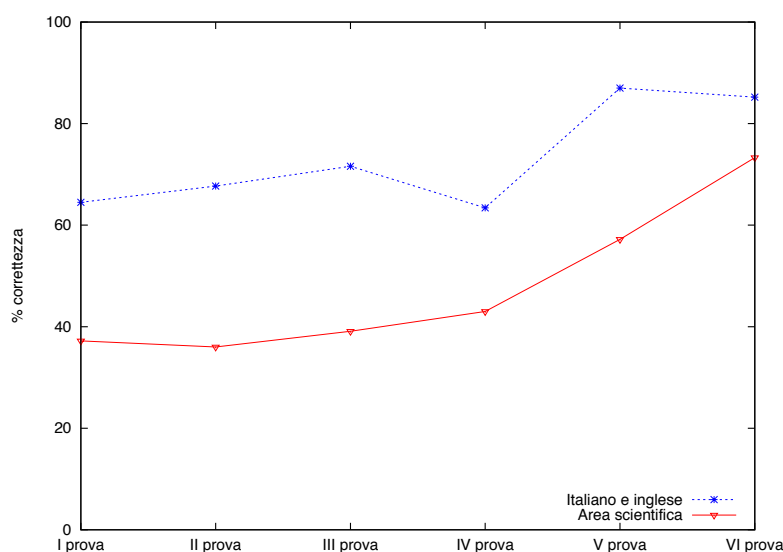


Figura 4.16: L'area scientifica e l'area linguistica nelle scuole secondarie di secondo grado

4.2 Le eccellenze

Il confronto descritto in questa sezione è scaturito da una prima intervista realizzata ad un'insegnante di una scuola secondaria di primo grado della provincia di Bologna. Nonostante le chiare finalità del progetto si è scoperto come alcuni istituti² abbiano selezionato i propri rappresentanti non tanto in base all'interesse, alla motivazione e ai margini di miglioramento, quanto ai successi scolastici e alla media voto. La diretta conseguenza di questo operato si traduce in risultati sicuri e alto prestigio e visibilità per l'istituto. Pur convenendo che a studenti bravi in tutte le materie scolastiche sia quasi sempre associato l'interesse per il ragionamento e la risoluzione di problemi in generale, in casi come questo si può affermare che il reale focus del progetto non sia stato colto pienamente. Considerato ciò, si è ipotizzato un appiattimento delle prestazioni di questi studenti *bravi* rispetto al resto dei partecipanti non soggetti a questa selezione. I risultati sembrano confermare quanto detto.

²Ulteriori interviste a istituti secondari di primo e secondo grado hanno confermato che non si è trattato di un caso isolato.

Si è deciso dunque di “spaccare” gli andamenti complessivi dei tre cicli di istruzione analizzando il comportamento dei soli studenti finalisti, le cosiddette eccellenze, separatamente fin dalla prima prova.³ Per quanto riguarda la scuola primaria, in Figura 4.17 è mostrato l’andamento degli studenti finalisti per le prime cinque prove, e in Figura 4.18 il confronto con tutti gli altri studenti. Le Figure 4.19, 4.20, 4.21 e 4.22 descrivono la situazione per la scuola secondaria di primo e secondo grado, rispettivamente. In termini assoluti, chiaramente, gli studenti che hanno disputato le finali hanno ottenuto un punteggio maggiore. In termini di miglioramento relativo, tuttavia, hanno avuto il rendimento minore in ognuno dei tre cicli di istruzione. Le eccellenze hanno mostrato un miglioramento evidente della loro prestazione: i *bravi* sono rimasti bravi; gli altri hanno mostrato un significativo aumento di competenze cognitive. Questo risultato è un campanello d’allarme per tutti quei dirigenti scolastici che, come negli esempi sopra citati, sacrificano gli studenti meno bravi, meno attivi, meno stimolati, ma potenzialmente in grado di appassionarsi al ragionamento logico e al problem solving migliorando le loro prestazioni in ogni campo, per un maggiore riscontro in termini di reputazione e merito.

4.3 Valutazioni generali dei risultati

Dall’indagine svolta non v’è alcun dubbio che tutti i ragazzi abbiano tratto beneficio dalle gare di problem solving. Non sempre il trend è stato positivo monotono crescente, ma in linea generale ci si può ritenere soddisfatti.

È curioso come gli stessi ragazzi, indipendentemente dal ciclo di istruzione, abbiano avuto un Δ di miglioramento maggiore durante il primo anno di sperimentazione. Senza distinguere quindi il ciclo di istruzione ma gli anni scolastici è possibile osservare che i due grafici abbiano pendenze diverse pur arrivando all’incirca sullo stesso punto. L’andamento del primo anno di sperimentazione presenta sempre un punto di partenza più basso rispetto a quello dell’anno successivo, i risultati finali invece tendono a convergere: se per il primo anno si arriva,

³Questo tipo di analisi, pur essendo cruciale, è stata possibile soltanto per le prove dell’anno scolastico 2010/2011, a causa della dimostrata l’impossibilità a reperire i dati degli anni precedenti.

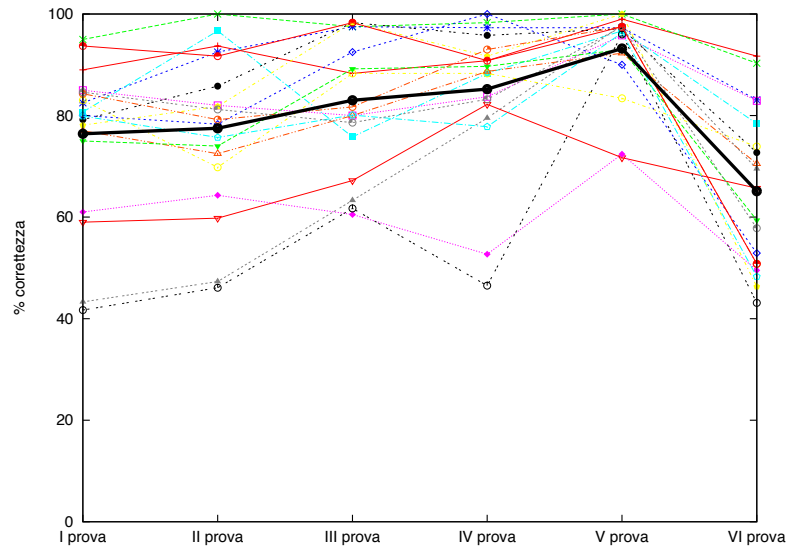


Figura 4.17: Scuola primaria, studenti finalisti

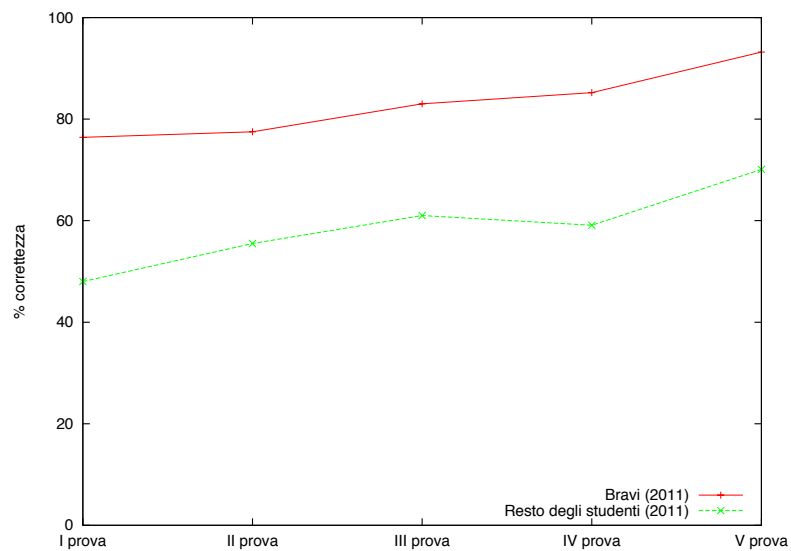


Figura 4.18: Scuola primaria: confronto fra “bravi” e il resto degli studenti

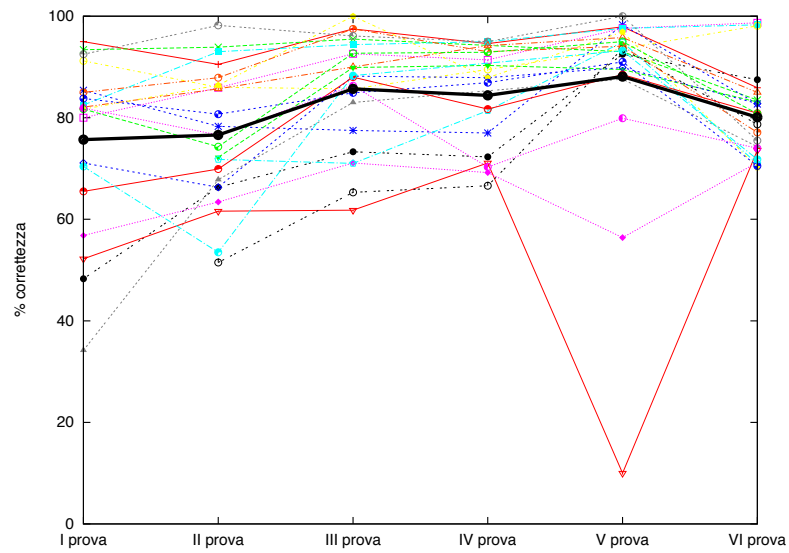


Figura 4.19: Scuola secondaria di primo grado: studenti finalisti

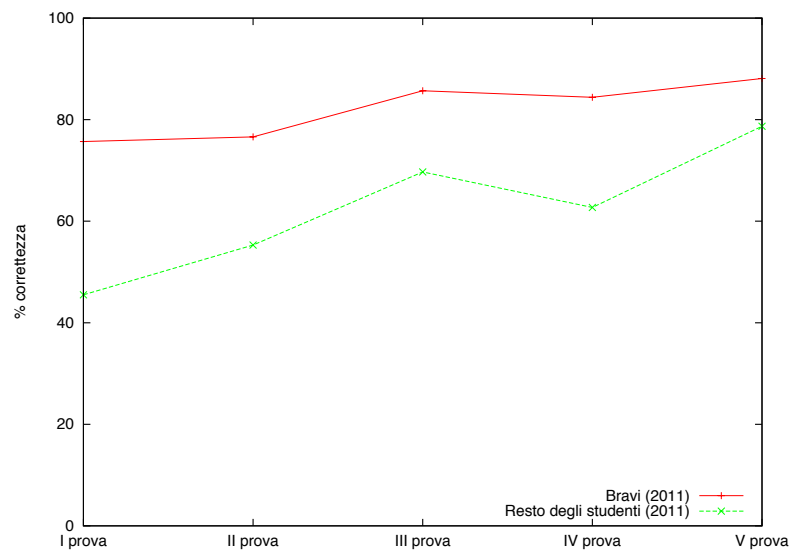


Figura 4.20: Scuola secondaria di primo grado: confronto fra “bravi” e il resto degli studenti

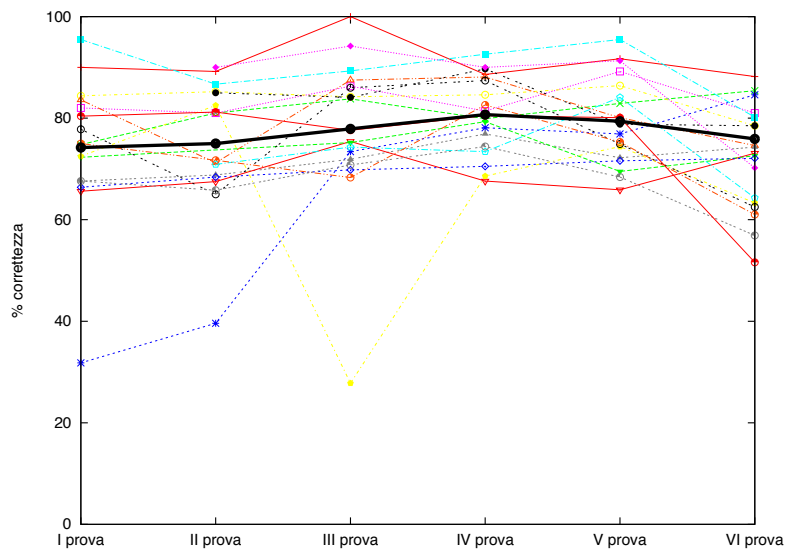


Figura 4.21: Scuola secondaria di secondo grado: studenti finalisti

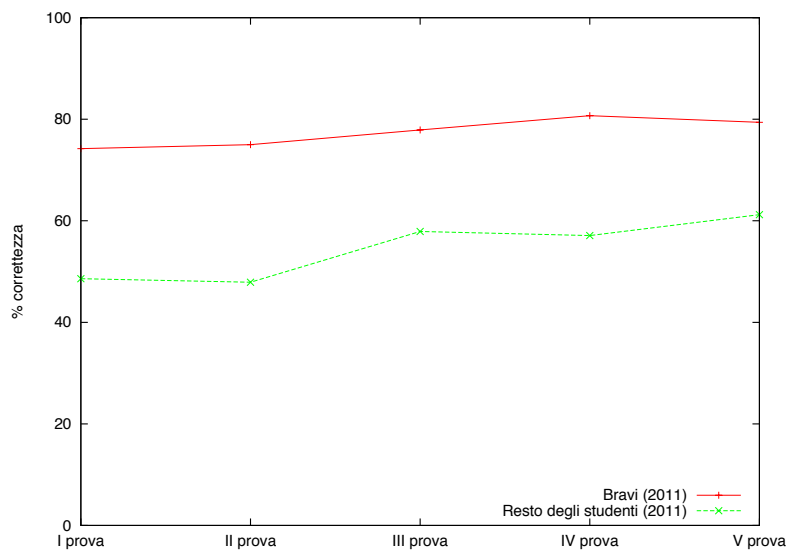


Figura 4.22: Scuola secondaria di secondo grado: confronto fra “bravi” e il resto degli studenti

prendendo ad esempio la scuola secondaria di primo grado, all'80% di correttezza, pare che proprio quel valore - circa - sia quasi come un learning upper bound per quei ragazzi. Pare che il miglioramento non sia costante variando la preparazione degli studenti.

Una causa di questo fenomeno potrebbe venire dagli insegnanti. Durante gli allenamenti e le gare gli insegnanti si trovano ad affrontare una serie di problemi - la maggior parte dei quali risultano nuovi anche per loro, alcuni conosciuti, altri no, alcuni più facili e altri per i quali gli studenti chiedono aiuto nella comprensione, nella risoluzione, nella standardizzazione. Se da una parte gli studenti imparano allenamento dopo allenamento, questo si riflette allo stesso modo sugli insegnanti. Dalle interviste si è scoperto come alcuni insegnanti, inizialmente sprovvisti e incapaci di produrre un aiuto, abbiano poi collaborato con soluzioni ad hoc di alcuni esercizi di matematica, grafi e knapsack problem. Si noti - cosa insolita per la categoria - che gli insegnanti stessi sono diventati parte integrante della sperimentazione.

Tuttavia questo fenomeno degli insegnanti che aiutano gli studenti nella preparazione delle gare introduce un ulteriore fenomeno. In molti, infatti, si saranno posti una domanda più che lecita: Gli studenti ottengono brillanti risultati perchè hanno svolto correttamente un problema che hanno capito e interiorizzato, oppure grazie all'applicazione meccanica di un procedimento che a loro risulta oltretutto poco chiaro come il problema stesso? Quest'argomento, chiamato *Teaching to The Test*, sarà discusso in una sezione successiva.

Dalle indagini sugli studenti *bravi*, come detto, si è scoperto che in questi primi anni di sperimentazione è stato in alcuni casi perso di vista l'obiettivo primario del progetto: far partecipare la maggior parte degli studenti, più o meno bravi, più o meno motivati, al fine di suscitare interesse e curiosità per una serie di attività dall'indiscusso potere formativo. Per fortuna però, l'essere venuti a conoscenza di questi differenti criteri di reclutamento ci ha permesso di indagare, appunto, sui suoi effetti. Come visibile dai plot, gli studenti non ammessi alle finali hanno ottenuto un Δ di miglioramento maggiore degli altri. Questo significa che le gare di problem solving hanno suscitato l'interesse e la curiosità di tutti i partecipanti, stimolando in particolar modo gli studenti che, durante le ore di lezione istituzionali, riportano un rendimento che ora si fatica definire inferiore a

gli altri - pur essendo vero, bensì inferiore alle loro stesse capacità.

Una piccola considerazione per i problemi somministrati. Per più di una categoria è emerso che gli stessi ragazzi che hanno ottenuto buoni risultati in una gara abbiano regredito esageratamente in quella successiva, e di nuovo migliorato in modo sospetto. Questo altalena senza significato trova l'unica spiegazione possibile nella difficoltà non uniforme, o non uniformemente crescente dei problemi. La scelta delle categorie con cui comporre le gare non è stata completamente coerente, sarebbe stato più saggio, ai fini di un'indagine, uniformarla in funzione di un'intera olimpiade.

4.4 Problemi dell'impianto e proposte

Durante e dopo l'indagine sono emersi una serie di problemi intrinseci e di altra natura, di una parte dei quali si sta già tenendo conto nell'anno di sperimentazione corrente. Oltre a prenderli in rassegna, saranno proposte minime modifiche all'impianto attuale.

Intrinseci Il fatto che le Olimpiadi di Problem Solving siano organizzate sotto forma di attività extrascolastiche le pone a rischio di una serie di considerazioni che, per la natura del progetto, rimangono irrisolvibili. Qualsiasi attività non obbligatoria, oltretutto non necessariamente svolta al mattino, è per definizione colta dalla minoranza. Emerge dunque il cosiddetto problema della *self selection*: gli studenti interessati, motivati e - in certi casi - con possibilità di essere a scuola fuori dal normale orario di lezione, parteciperanno; gli altri non parteciperanno.

Senza indagare sulle motivazioni, il progetto non prevede nessun tipo di preparazione preliminare degli insegnanti. La mancanza di cattedre di informatica nelle scuole dell'obbligo fa sì che una parte degli insegnanti, specialmente nella scuola primaria e secondaria di primo grado, alla vista dei quesiti rimanga disorientata e non sia in grado di preparare gli studenti alle competizioni. Soltanto i più lungimiranti si armano di volontà e pazienza sacrificando tempo alla propria vita privata per studiare i problemi presentandone delle alternative su cui lavorare.

A tal proposito è doveroso riportare un'intervista ad un'insegnante di italiano di un istituto comprensivo della provincia di Ravenna riguardo alla propria esperienza di partecipazione alle gare di problem solving.

“... Pur insegnando italiano, pur non conoscendo bene le altre materie, ho proposto attività di problem solving ai miei bimbi fin dalla prima classe, seguendo uno strano istinto che mi diceva di procedere in tal senso, anche se era molto frustrante accorgermi di “perdere” molto tempo nella preparazione delle attività che oltre tutto c’entravano pressoché nulla col mio ambito e vi sottraevano tempo ed energie. Non sapevo cosa avrei raccolto, forse al limite ne poteva trarre vantaggi - in termini di sviluppo della logica - il mio collega di matematica, estremamente restio nell’uscire dal tracciato del suo percorso tradizionale e che quindi non ne ha mai voluto sapere, pur lasciandomi libera di invadere il suo spazio.”

Dalle parole di quest’insegnante emerge tutta la frustrazione di chi non si sente in grado di affrontare il compito che gli è stato affidato. Di questa incolpevole incapacità si dovrebbe prendere nota, con sufficiente attenzione. Continua:

“Già dalla prima e tanto di più in seconda, però, mi accorgevo che i bambini mettevano un’attenzione particolare nella lettura dei quesiti, facendo attenzione alle singole parole che di solito si leggono distrattamente, ma che qui, si rendevano conto, incidevano radicalmente sulla buona riuscita dell’esercizio: *almeno, tutti, la metà, possibile, probabile, improbabile, non, ...* (anche perchè preparavo dei mini tornei a squadre con tanto di premio e gli esercizi erano al computer, realizzati con un software per la creazione di test che riporta al termine il risultato, il tempo impiegato, etc. . . ; erano insomma in gara con se stessi e con gli altri al contempo). In generale, poi, mi sembrava di avere dei bimbi particolarmente “svegli”, organizzati, precocemente autonomi nella gestione pratica delle loro cose e dell’aula. Ad esempio usano e gestiscono il diario e l’organizzazione dei compiti e preparazione dello zaino con i materiali richiesti per il giorno dopo fin dalla fine della prima classe, cosa assolutamente insolita, in quanto la totalità dei miei

colleghi ne introduce timidamente l'utilizzo in terza. Quest'anno, iniziando a insegnare anche in una quarta, con bimbi più grandi, ne ho avuto la conferma, in qualche modo. I miei bimbi di terza sono di fatto molto più "grandi", e questo lo riconosce anche il mio collega di matematica che dice di aver anticipato, coi piccoli, parecchi argomenti che normalmente affrontava nelle annualità successive. Un esempio pratico: le olimpiadi di problem solving. Ho iscritto le quattro squadre formate dagli alunni di quarta, ma il giorno della competizione - pur avendo selezionato per loro solo 4 o 5 items - non riuscivano a suddividersi i compiti proficuamente. A gestire il lavoro di gruppo, insomma. Ho chiamato allora otto bimbi di terza e ne ho affiancati due per ogni squadra. Hanno preso loro in mano la situazione, ripartendo gli incarichi e occupandosi in prima persona della raccolta finale delle risposte e dell'immissione dei dati al computer."⁴

Di organizzazione Durante l'indagine sono emerse diverse complicazioni dovute a una cattiva gestione del database del progetto. Sono stati necessari parecchi confronti incrociati per stabilire quali dati fossero da considerare attendibili fra numerosi duplicati. Trascurando di chiedersi dell'esigenza - da parte del gestore - di fornire i dati sotto forma di tabelle html, e per quale motivo la maggior parte dei file avesse uno o più duplicati diversi, si è riusciti a effettuare la dovuta scrematura in modo da poter operare su materiale corretto e univoco.

Per ricondursi alla questione delle categorie, invece, si ritiene sufficientemente adeguata la suddivisione dei problemi. Tuttavia, qualora lo si ritenesse necessario, sarà sempre possibile ampliare lo spettro di esercizi e attività [21]. Come accennato in precedenza, però, mancando qualche occorrenza di qualche categoria alla lunga delle sei prove annue, diviene più problematico tracciare un andamento relativo, appunto, alle singole categorie. Ai fini statistici è quindi opportuno comporre le prove di almeno un'occorrenza di ogni categoria. In questo caso avranno un ruolo cruciale gli ideatori dei problemi, poichè sarà necessario crearne un vasto

⁴Tutte le altre interviste effettuate in scuole primarie e secondarie di primo e secondo grado sono sulla falsariga di quella riportata.

insieme per ogni categoria, al fine di rendere le prove meno standardizzate, meno meccaniche e dare spazio allo sviluppo di tutte le abilità cognitive dei partecipanti.

Benchè la suddivisione possa ritenersi accettabile, le interviste agli insegnanti suggeriscono modifiche al livello di difficoltà degli esercizi adattandolo al ciclo di istruzione e all'anno scolastico. Si potrà iniziare con esercizi molto semplici occultando via via le informazioni utili e inserendone altre fuorvianti, per poi passare ad esercizi concettualmente più complessi. Benchè questa modifica risulti più significativa con una partecipazione più consistente e coerente delle scuole, si avrà quindi un'ulteriore suddivisione dei problemi all'interno di ogni categoria.

Un'indagine in progetto - purtroppo non realizzata - avrebbe voluto il confronto fra i risultati delle olimpiadi e i test Invalsi⁵, con l'obiettivo di individuare, con granularità "squadra", unica possibile, l'eventuale miglioramento dei gruppi di studenti partecipanti alle gare nei tradizionali argomenti di curriculum. Questioni legate alle tempistiche burocratiche non ci hanno permesso di reperire i dati dei test Invalsi relativi agli istituti scelti come campione. Poichè si ritiene di primaria importanza un confronto con tali test, e contenendo questi informazioni sui singoli studenti di ogni istituto, è necessario preparare il terreno frammentando ulteriormente l'unità di confronto da "squadra" a "singolo componente". Questo livello di dettaglio potrebbe ispirare un maggior numero di indagini, una su tutte la percentuale di partecipazione di maschi e femmine e relative differenze di prestazioni. Per le prossime edizioni il database, pur continuando a raccogliere le informazioni di ogni squadra, necessarie alle valutazioni interne, dovrebbe quindi essere in grado di distinguere quei problemi che prevedono risoluzioni individuali.

Altre considerazioni

Teaching to the test Il teaching to the test ha ottenuto crescente considerazione attraverso lo studio dei risultati dei test standardizzati a risposta multipla. Pur non essendo le prove delle olimpiadi di problem solving strutturate in questo modo, si è potuto constatare indifferentemente l'occorrere di questo fenomeno.

⁵Confronto possibile soltanto per le categorie di italiano e matematica, per allineamento ai test Invalsi.

Il compito di ogni insegnante è prima di tutto preoccuparsi dell'apprendimento dei suoi studenti. Idealmente, gli studenti saranno capaci di sviluppare le abilità necessarie ad interiorizzare ciò che hanno imparato e ad applicarlo in nuove situazioni. In questo caso l'insegnante avrà favorito un apprendimento vero e proprio all'interno della sua classe. L'avvento dei test nazionali di valutazione (vedi Invalsi) come tutte le forme di verifica test-like, hanno in parte interferito con questi obiettivi, avendo come scopo la misurazione della conoscenza degli studenti e la relativa calibrazione della loro istruzione. Questi test standardizzati sono ritenuti benchmarks di successo, spesso (mal)utilizzati per fare confronti fra studenti, scuole, corpi amministrativi, con il risultato di avere insegnanti e scuole spesso biasimati per i loro scarsi risultati. I test standardizzati, quando usati appropriatamente, si dimostrano un aiuto per gli insegnanti nel trovare carenze e punti di forza dei propri studenti [22]. Tuttavia, data l'importanza di queste valutazioni, spesso si cade nell'errore di sacrificare intere parti di curriculum a favore di argomenti ad "alta priorità" e di dedicare molto tempo alla preparazione dei test. Tale preparazione, escludendo gli insegnanti di alto livello, è di solito costituita esattamente da copie perfette degli esercizi, e molto raramente si ripercuote in positivo sull'apprendimento. Questo meccanismo risulta essere nocivo per gli studenti e altamente frustrante per gli insegnanti [23].

Preparazione degli insegnanti Si capisce bene come il ruolo dell'insegnante sia cruciale in questi casi. Un'adeguata nonché intelligente preparazione ai test preserva gli studenti dall'imparare meccanicamente a risolvere gli esercizi, stimolando il ragionamento e l'applicazione dei concetti a un ampio spettro di situazioni diverse. Purtroppo però, la situazione attuale in Italia non è delle migliori. L'alto tasso di precariato, la diversa motivazione, la diversa preparazione e la totale assenza di formazione preliminare degli insegnanti rendono le cose più difficili e lontane da quello che si può chiamare un background ottimale per una sperimentazione.

Capitolo 5

Conclusioni

5.1 Conclusioni

L'Informatica ha cambiato la nostra vita. Nella comunicazione, nella scrittura, nella lettura, nell'ascolto della musica, radio e televisione, nella fotografia, nella ricerca di informazioni, negli affari e molto altro. Ma la percezione più comune dell'Informatica non è quella di una scienza vera e propria, ma quella di un servizio. Un servizio posto esattamente allo stesso livello dei servizi che essa stessa è in grado di produrre. In molti confondono l'Informatica con il programmare. Non sono coscienti della complessità, dell'utilità, ma spesso anche soltanto dell'esistenza, dei contenuti scientifico-culturali che sono alla base di tutti quei servizi che oggi ci rendono la vita così semplice e che sarebbe in grado di trasmettere. C'è anche chi pensa che le scoperte più rilevanti in ambito informatico siano già arrivate e che il potenziale sviluppo rimanga soltanto nell'ingegnerizzazione. L'Informatica e le tecnologie che essa sviluppa sono il cuore della nostra economia, della nostra vita di tutti i giorni e delle imprese. L'era digitale ha trasformato il mondo e la forza lavoro.

Tuttavia l'insegnamento dell'Informatica nelle scuole dell'obbligo fallisce [24]. L'educazione scolastica sta ignorando tristemente la preparazione degli studenti sui principi fondamentali dell'Informatica che sono necessari per il successo futuro. Per sapersi districare in un mondo sempre più computing intensive e per essere preparati ai lavori del XXI secolo, gli studenti devono avere una comprensione più profonda dei principi dell'Informatica.

Negli ultimi anni sta maturando un nuovo contesto culturale in cui presentare l'Informatica come disciplina: il Computational Thinking. Il computational thinking è una visione dall'alto che serve a guidare insegnanti, genitori, ricercatori e corpi scolastici amministrativi su come agire per cambiare l'immagine di questo settore nella società. Deve in modo particolare raggiungere la scuola dell'obbligo e mandare due messaggi importanti: i problemi scientifici intellettualmente stimolanti e affascinanti non finiscono mai, e per trovarli è necessario allargare i limiti della nostra creatività; ci si può specializzare in informatica e fare qualsiasi cosa: iniziare una carriera in medicina, in legge, in economia, in politica, o qualsiasi tipo di disciplina scientifica e ingegneristica; anche nel campo delle arti.

Con un sistema scolastico così conservativo come quello italiano modifiche al curriculum sono difficili da mettere in pratica; l'istanziamento di ulteriori fondi, di cattedre di informatica, o l'organizzazione di corsi di formazione sono richieste che andranno presentate in tempi più sereni. Per questi motivi ed in questo contesto il Ministero ha individuato nelle Olimpiadi di Problem Solving una maniera per sensibilizzare le istituzioni scolastiche partendo "dal basso". Da tre anni organizza queste attività extrascolastiche su tutti i cicli d'istruzione della scuola dell'obbligo con allenamenti, prove di qualificazione e fasi finali che hanno lo scopo di trovare delle rilevanze nello sviluppo delle conoscenze e avere un criterio oggettivo per far emergere le eccellenze. I risultati della sperimentazione sono incoraggianti e sembrano confermare la giusta direzione. Si sono registrati miglioramenti da parte di tutti i ragazzi ed in particolare su classi di problemi che non avevano mai affrontato finora come il knapsack, i grafi e il project management. Un risultato importante è stato anche l'ottimo miglioramento relativo degli studenti medi, ovvero con medie voto (nelle discipline istituzionali) inferiori rispetto alle eccellenze.

In definitiva, il progetto Problem Solving si sta rivelando assai utile a chi partecipa, insegnanti compresi, che come si è scoperto sono indirettamente coinvolti nella sperimentazione. Restano tuttavia diversi problemi, alcuni dei quali difficilmente risolvibili (vedi problemi intrinseci). Oltre ad una messa a punto generale dell'impianto di gestione del progetto, si potrebbe ragionare sul problema della self selection che, trattandosi di attività extrascolastiche, non sarà destinato a sparire.

5.2 Sviluppi futuri

Una sfida fondamentale dal punto di vista educativo è rappresentata dalla necessità di allargare la partecipazione al maggior numero di studenti, di sesso maschile e femminile in egual misura. La corrente di pensiero attualmente in voga fra i ricercatori presenta un modello di insegnamento diverso da quello tradizionale, che attraverso la progettazione e programmazione di giochi (il cosiddetto *game design*) introduce agli studenti il *computational thinking* [25]. Il *game design* combina il modello psicologico del flusso¹ (Flow psychological model) [26], il framework delle competenze FIT [27], e un ambiente di programmazione visuale. Gli aspetti motivazionali ed educativi dello sviluppo di semplici giochi sono difficili da bilanciare, tant'è che spesso sacrificano gli obiettivi puramente didattici. L'utilizzo dei cosiddetti *visual programming languages* rende l'Informatica accessibile ad un numero maggiore di studenti rispetto ai metodi tradizionali, poichè riesce a catturarne l'interesse. Questi linguaggi visuali, di cui AgentSheet, Alice e Scratch² (Figura 5.1) sono degli esempi, possono risultare effettivi nell'insegnamento della progettazione, della programmazione e del *computational thinking*. Gli obiettivi di questo modello sono principalmente tre:

- i) Capire la connessione fra motivazione e apprendimento
- ii) Essere in grado di concettualizzare le competenze
- iii) Supportare la progettazione e lo sviluppo con la tecnologia [28]

Spostando il focus, dunque, dalla programmazione alla progettazione, molti principi di *computational thinking* possono essere più facilmente trasmessi. Iniziando dallo studio del problema, i test dei casi limite e la progettazione di una soluzione ottimizzata, ben strutturata e facilmente modificabile si impareranno concetti come la decomposizione, l'astrazione e il riconoscimento di pattern.

¹L'obiettivo è far rimanere lo studente nella zona di flusso ottimale il più tempo possibile senza sfociare nell'ansia o nella noia, e per aiutarlo l'insegnante ha a disposizione varie forme di aiuto quali un'esplicita istruzione data sul momento, l'interazione con altri studenti o curricula progettati in previsione dei problemi successivi.

²<http://scratch.mit.edu/>

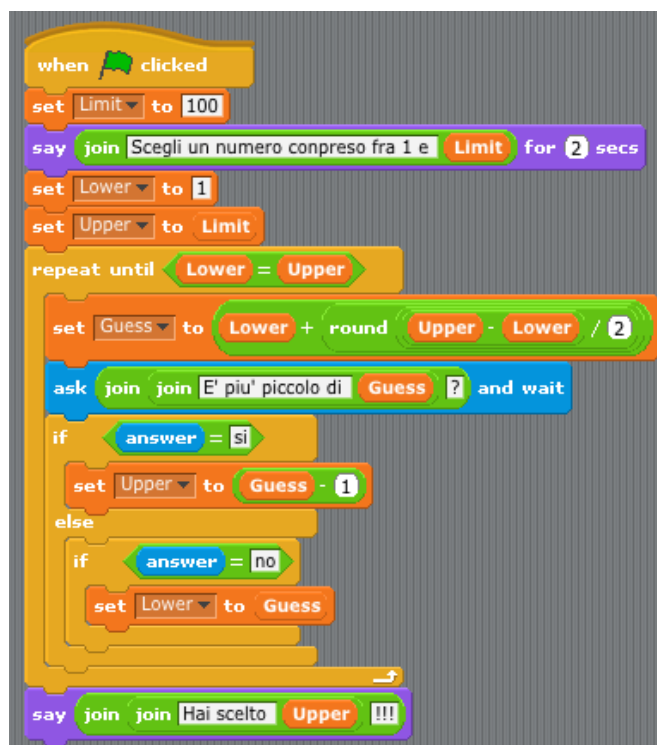


Figura 5.1: Linguaggio visuale Scratch: esempio di ricerca binaria

Per valutare in modo preciso che cosa lo studente effettivamente impara, alcuni ricercatori hanno costruito un sistema di riconoscimento di pattern di computational thinking essenzialmente basato sul confronto fra i lavori degli studenti e una serie di programmi di riferimento [29]. In questo modo si capirà quando gli studenti acquisteranno familiarità con l'uso di determinati pattern, e vi sarà la possibilità di offrire esercitazioni ad hoc ad ogni singolo studente.

La capacità di ridurre i problemi di sintassi e lo spirito del Web 2.0 per poter condividere i propri progetti con insegnanti e compagni di scuola rendono questo approccio "pedagogico" all'insegnamento dell'Informatica un affascinante mondo ricco di sfide educative e allo stesso tempo divertenti e stimolanti, e che completeranno la base di conoscenza necessaria per i ragazzi che dovranno formare il futuro della società.

Bibliografia

- [1] J. M. Wing. “Computational Thinking”. In: *Communications of the ACM* 49.3 (2006).
- [2] J. M. Wing. “Computational Thinking, What and Why”. In: *The Link* (2011).
- [3] J. M. Wing. “Computational thinking and thinking about computing”. In: *Phil. Trans. R. Soc. A* (2008).
- [4] R. C. Anderson, R. J. Spiro e M. C. Anderson. *Schemata as scaffolding for the representation of information in connected discourse*. Rapp. tecn. University of Illinois, 1977.
- [5] H. Gardner. “The Minds New Science: A History of the Cognitive Science Revolution”. In: *Basic Books* (1987).
- [6] L. Perkovic e A. Settle. *Computational Thinking across the Curriculum: A Conceptual Framework*. Rapp. tecn. 13. DePaul University, 2010.
- [7] P. J. Denning. *Great principles of computing*. URL: <http://greatprinciples.org>.
- [8] Google. *Exploring Computational Thinking*. 2010. URL: <http://www.google.com/edu/computational-thinking/>.
- [9] President’s Information Technology Advisory Council. *Computational Science: Ensuring America’s Competitiveness. Report to the President*. Rapp. tecn. Executive Office of the President of the United States, 2005.
- [10] K. G. Wilson. *The renormalization group and critical phenomena*. Nobel lecture. 1982.

- [11] B. Chazelle. “The Convergence of Bird Flockin”. In: *ACM-SIAM Symposium on Discrete Algorithms* (2009).
- [12] D. Abraham, A. Blum e T. Sandholm. “Clearing algorithms for barter exchange markets: enabling nationwide kidney exchanges”. In: *ACM Conference on Electronic Commerce (EC)*. New York, 2007, pp. 295–304.
- [13] Carnegie Mellon University. *Machine Learning Department 2008 Research projects home page*. 2008. URL: <http://www.ml.cmu.edu/research/index.html>.
- [14] J. Fisher e T Henzinger. “Executable cell biology”. In: *Nature Biotechnology* (2007), pp. 1239–1249.
- [15] R. E. Bryant, K. Sutner e M. Stehlik. *Introductory Computer Science Education at Carnegie Mellon University: A Deans’ Perspective*. Rapp. tecn. Carnegie Mellon University, 2010.
- [16] National Research Council Committee for the Workshops on Computational Thinking. *Report of a Workshop on the Scope and Nature of Computational Thinking*. 2010.
- [17] P. J. Denning. “Beyond computational thinking”. In: *Communications of the ACM* 52.6 (2009).
- [18] S. Martini. “Elogio di Babele”. In: *Mondo digitale 2* (2008).
- [19] Ministero dell’Istruzione dell’Università e della Ricerca. *Syllabus di Elementi di Informatica per la scuola dell’obbligo*. 2010. URL: www.istruzione.it.
- [20] Ministero dell’Istruzione dell’Università e della Ricerca. *Linee guida per il passaggio al nuovo ordinamento II. PP.e II. TT*. 2010. URL: www.istruzione.it.
- [21] P. Phillips. *Computational Thinking, a problem-solving tool for every classroom*. NECC, Atlanta. 2007.
- [22] J. H. McMillan. “Fundamental Assessment Principles for Teachers and School Administrators”. In: *Practical Assessment, Research & Evaluation* (2006).

- [23] L. Volante. “Teaching to the test: what every educator and policy-maker should know”. In: *Canadian Journal of Educational Administration and Policy* 35 (2004).
- [24] C. Wilson et al. *Running on Empty: The Failure to Teach K-12 Computer Science in the Digital Age*. Rapp. tecn. ACM CSTA, 2010.
- [25] A. Reppenning e A. Ioannidou. *Broadening participation through scalable game design*. Rapp. tecn. University of Colorado, 2008.
- [26] M. Csikszentmihalyi. *Flow: The psychology of Optimal Experience*. New York: Harper Collins Publisher, 1990.
- [27] Committee on Information Technology Literacy e National Research Council. *Being Fluent with Information Technology*. Washington, D.C.: National Academy Press, 1999.
- [28] R Rose et al. “Computational Thinking, Computational Science and High Performance Computing in K-12 Education: White Paper on 21st Century Education”. In: *Educational technology & change journal* (2011).
- [29] SIGCSE ’11 Proceedings of the 42nd ACM technical symposium on Computer science education, cur. *Recognizing computational thinking patterns*. 2011.
- [30] *Sito delle Olimpiadi di Problem Solving*. URL: <http://www.olimpiadiproblemsolving.it>.
- [31] L. Bond. *Teaching to the Test. A commentary on the thorny issue of high-stakes testing and the pressures on teachers to teach to the test*. Rapp. tecn. Carnegie Foundation, 2004.
- [32] A. V. Dam, J. Foley, J. Gutta et al. *Creating Environments for Computational Researcher Education*. Rapp. tecn. Computing Research Association, 2010.
- [33] K. H. Koh et al. *Towards the Automatic Recognition of Computational Thinking for Adaptive Visual Language Learning*. Rapp. tecn. University of Colorado, 2010.

- [34] ITiCSE 10 Proceedings of the fifteenth annual conference on Innovation e technology in computer science education, cur. *Using scalable game design to teach computer science from middle school to graduate schools*. 2010.
- [35] D. Moursund e D. Ricketts. *Computational thinking*. 2011. URL: http://iae-pedia.org/Computational_Thinking.
- [36] Wing, Cuny e Snyder. *Center for computational thinking*. 2006. URL: <http://www.cs.cmu.edu/~CompThink>.