

## 7. La teoria degli algoritmi nella scuola secondaria di secondo grado

*Luca Forlizzi<sup>1</sup>, Guido Proietti<sup>2</sup>*

Grande è l'insoddisfazione della comunità accademica internazionale che si occupa di didattica dell'informatica nei riguardi dello stato in cui versa l'insegnamento della disciplina nelle scuole. Si ritiene infatti che non si insista a sufficienza sui fondamenti scientifici della disciplina, e che molto spesso, trascurando financo importanti contenuti tecnologici, ci si riduca a mero addestramento all'utilizzo delle tecnologie ICT, l'aspetto più superficiale della cosiddetta *Digital Literacy*. Questa situazione viene denunciata sia in numerosi articoli scientifici [2], [9], [7], [10], sia in report di organizzazioni accademiche [6], [1]. Questa situazione sembra comune un po' in tutto il mondo, tranne in alcune nazioni che si distinguono per aver intrapreso riforme dei curricula informatici fin dalla scuola primaria [4].

Nel caso italiano, cambiare questo stato di cose richiede sia un adeguamento delle indicazioni ministeriali sui programmi di insegnamento, sia una diversa formazione dei futuri docenti di informatica. La recente istituzione di percorsi universitari abilitanti all'insegnamento dell'informatica nella scuola secondaria di secondo grado, ovvero i TFA e i PAS, costituisce una preziosa occasione per la comunità accademica italiana per intervenire nella definizione delle competenze dei futuri docenti.

In questo lavoro prendiamo in esame l'insegnamento di due pilastri caratterizzanti dell'informatica: la teoria della calcolabilità e quella

---

<sup>1</sup> Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica, Università degli Studi dell'Aquila.

<sup>2</sup> Dipartimento di Informatica, Università degli Studi dell'Aquila e Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", Consiglio Nazionale delle Ricerche.

degli algoritmi. In particolare, partendo da una riflessione sulla percezione culturale dell'informatica, descriviamo l'approccio che abbiamo utilizzato al fine di consapevolizzare i futuri docenti in merito al ruolo formativo peculiare che l'informatica in generale, e l'algoritmica in particolare, può svolgere sugli studenti delle scuole secondarie di secondo grado. Infine, presentiamo i risultati di un'indagine di gradimento del corso che abbiamo condotto tra i discenti dei corsi TFA e PAS, i cui risultati si sono e si stanno dimostrando utili per i successivi affinamenti dei contenuti proposti.

## 7.1. Algoritmi nelle scuole: aspirazioni e realtà

Gli algoritmi sono naturalmente uno dei concetti fondanti della disciplina e non sorprende che il loro studio sia centrale in numerose proposte di curricula e sperimentazioni didattiche [8], [3], [5], [7]. Nelle attuali indicazioni ministeriali essi sono sì presenti, pur tuttavia non con la centralità loro dovuta.

La nostra esperienza di confronto con i docenti della scuola ci suggerisce che gli algoritmi vengono in genere considerati come una sorta di esercizio propedeutico alla programmazione. Lo studio degli algoritmi viene spesso inteso come:

- la formalizzazione dei diagrammi di flusso
- l'esposizione di alcuni esempi di algoritmi notevoli (ordinamento su tutti) il cui scopo unico è veicolare agli allievi l'idea che prima di cominciare a scrivere codice devono fermarsi a pensare.

Nulla o poco si dice quindi in merito al significato più profondo di algoritmo come vero e proprio idioma di una scienza a tutto tondo, ancorché moderna, ovvero l'informatica. A nostro avviso, tale distorsione ha radici antiche, che rimandano addirittura alla radice linguistica del termine informatica. Tale termine deriva infatti dal tedesco *Informatik*, contrazione di *Automatische Informationsverarbeitung*. Nell'accezione italiana, dunque, il termine viene associato sia alla pratica di utilizzo di dispositivi automatici di elaborazione dati (computer, in senso lato: PC, tablet, smartphone, etc.), sia alla ben più profonda attività legata alla loro *programmazione*, vero e proprio connubio di ingegno e tecnica, l'educazione alla quale richiede un approccio pedagogico sistematico e di carattere scientifico. Si noti invece che nella

lingua inglese tale dicotomia è esplicitata nel lessico, in quanto il termine informatica si traduce con:

- *Information and Communication Technology (ICT)*, quando ci si riferisce ai dispositivi tecnologici e al loro utilizzo pratico;
- *Computer Science (CS)*, quando ci si riferisce allo studio sistematico dei processi computazionali (e quindi ai principi di programmazione di un elaboratore).

La prima declinazione rimanda quindi all'*empirismo* in filosofia: l'esperienza (cioè l'uso dello strumento) è alla radice della conoscenza. È questo l'aspetto che tuttora viene privilegiato nei corsi di formazione continua e in quelli scolastici primari di alfabetizzazione informatica in Italia (locuzione espressamente utilizzata nella Riforma Moratti del 2003): il discente viene istruito al mero uso di alcune delle applicazioni digitali di largo consumo, con ampie distorsioni di natura commerciale (in alcune circolari ministeriali viene addirittura esplicitamente previsto l'addestramento all'utilizzo di prodotti proprietari!). Tale approccio ha diverse implicazioni negative, tra le quali citiamo:

- l'incapacità di utilizzo consapevole dei mezzi di calcolo, ovvero mancanza di una vera cultura (o educazione) digitale, la quale consente invece di padroneggiare gli strumenti tecnologici e di muoversi a proprio agio nella comunità virtuale (la rete);
- la rapida obsolescenza delle informazioni acquisite, stante il livello di continua innovazione (hardware e software) in ambito tecnologico.

La seconda declinazione del termine inglese rimanda invece al *razionalismo* in filosofia: la ragione pura (conoscenza *a priori*) prevale sull'esperienza sensoriale (conoscenza *a posteriori*). In questo senso, la programmazione diventa quindi assimilabile ad un'attività matematica, con le tutte le implicazioni di carattere epistemologico che ne conseguono.

L'informatica con quest'ultima accezione è quindi, a nostro avviso, una disciplina scientifica pervasiva e con caratteristiche peculiari, che però stenta a trovare un suo ruolo nel panorama scolastico italiano (e mondiale), anche per via dei richiamati dissensi sulla sua identità culturale.

Nella comunità accademica, c'è invece ampio consenso riguardo al fatto che l'obiettivo principale dell'informatica sia quello di definire degli strumenti metodologici atti a sviluppare, sul piano psicologico, comportamentale ed operativo, le abilità necessarie a risolvere efficientemente ed automaticamente problemi computazionali per i quali non si

possiede a priori una procedura risolutiva. Questo approccio conduce quindi allo sviluppo nel discente del cosiddetto *pensiero computazionale*, che completa e complementa il classico sviluppo del pensiero logico-matematico. Il pensiero computazionale è un *modus cogitandi* finalizzato ad automatizzare la risoluzione di un dato problema mediante la definizione di una *soluzione algoritmica*, ovvero una sequenza accuratamente descritta di passi, ognuno dei quali appartenente ad un catalogo ben definito di operazioni di base ammissibili. L'algoritmo diventa quindi, a tutti gli effetti, il concetto centrale attorno al quale ruotano tutti gli aspetti fondanti dell'informatica intesa come *computer science*.

## 7.2. Teoria della computabilità per i futuri docenti di informatica

L'obiettivo fondamentale dei percorsi TFA e PAS che abbiamo costruito è stato proprio quello di aumentare la sensibilità dei futuri docenti rispetto all'importanza dello sviluppo del pensiero computazionale nei discenti delle scuole. Per costruire dei sillabi coerenti con le premesse, ci siamo ispirati all'insegnamento di due numi tutelari dell'informatica:

- “Il ragionamento matematico può essere considerato piuttosto schematicamente come l'esercizio di una combinazione di due capacità, che possiamo chiamare intuizione e ingegnosità.” (Alan Turing (1912-1954))
- “Se è vero che un problema non si capisce a fondo finché non lo si deve insegnare a qualcun altro, a maggior ragione nulla deve essere compreso in modo più approfondito di ciò che si deve insegnare ad una macchina, ovvero di ciò che va espresso tramite un algoritmo.” (Donald Knuth, autore di *The Art of Computer Programming*).

Per risolvere un problema computazionale, bisogna quindi coltivare e sviluppare l'intuito e l'ingegno del discente attraverso la comprensione iniziale della natura del problema stesso (ovvero del suo nucleo matematico), seguita poi dalla progettazione di una appropriata procedura di risoluzione algoritmica (laddove possibile!), ovvero una sequenza di istruzioni formali che potranno poi essere tradotte in un linguaggio intellegibile al computer (programma), il tutto concluso da una meticolosa verifica della correttezza del risultato.

Abbiamo dunque ritenuto opportuno, dedicare, sia nel TFA che nel PAS, un corso specifico sull'anima algoritmica dell'informatica, o più propriamente sulla *teoria della computabilità*, che a sua volta può essere suddivisa in due grandi filoni:

- La *teoria della calcolabilità*, ovvero lo studio della (ir)risolubilità dei problemi computazionali mediante un procedimento algoritmico;
- La *teoria degli algoritmi e della complessità computazionale*, ovvero lo studio delle risorse di calcolo (principalmente tempo di esecuzione e spazio di memoria utilizzato) necessarie e sufficienti ad un algoritmo (esatto, approssimato, randomizzato) per risolvere un problema computazionale.

Nella prima parte ci si poneva dunque un obiettivo ambizioso: trasmettere integralmente il significato di *calcolabilità*, tuttora relegato ai corsi di livello universitario. A nostro avviso si tratta invece di un concetto imprescindibile nel momento stesso in cui si introduce il principio della risoluzione *automatica* (o *algoritmica*) di un problema. Per inciso, si ritiene che la presentazione del paradosso insito nel *problema dell'arresto*, utilizzato allo scopo di mostrare l'esistenza di problemi non risolubili algebricamente, sia pienamente trasferibile ai discenti della scuola secondaria di secondo grado.

La seconda parte sulla teoria degli algoritmi e della complessità computazionale si poneva invece obiettivi ugualmente ambiziosi, ma diversificati: da un lato, molti dei concetti esposti erano pensati per essere trasferiti ai discenti, dall'altro lato alcuni contenuti sono stati pensati soprattutto in veste di arricchimento culturale dei docenti, al fine di sensibilizzarli circa l'importanza di trattare questi argomenti nella scuola e di costituire quell'*humus* propizio allo sviluppo di competenze più avanzate in materia.

Il tutto è stato illustrato cercando di utilizzare un linguaggio rigoroso, ma senza eccedere nel formalismo, con l'obiettivo quindi di fornire delle idee e del materiale da riutilizzare in classe (opportunamente adattato alle esigenze di ciascuno). Concretamente, tale approccio ha portato allo sviluppo di tre tematiche principali, le prime due delle quali riutilizzabili in classe, mentre la terza a carattere prettamente divulgativo e formativo verso i docenti, ovvero:

1. Facile, difficile, impossibile: spunti di teoria della calcolabilità e principali classi di complessità computazionale dei problemi.
  - Che cos'è un algoritmo?
  - È sempre possibile risolvere (algoritmicamente) un dato problema? Il problema dell'arresto.
  - Caratterizzazioni dei problemi in funzione della loro "difficoltà" computazionale. Macchine di Turing deterministiche e non deterministiche. Le classi di complessità P, NP, EXPTIME.
2. Essere algoritmista: progettare un algoritmo corretto, efficiente, e possibilmente ottimo.
  - Un esempio istruttivo: il problema dell'ordinamento.
  - Algoritmi di ordinamento elementari a confronto: prestazioni nel caso migliore, medio e peggiore.
  - Quanto è difficile ordinare? La delimitazione inferiore alla complessità computazionale di un problema.
  - Algoritmi di ordinamento ottimi.
3. Quando il problema è troppo arduo e tutto il resto fallisce: gli algoritmi di approssimazione e il potere della randomizzazione.
  - Un altro modo di definire la classe NP: il concetto di certificato.
  - La classe dei problemi NP-completi.
  - Il problema da 1 Milione di Dollari: P versus NP.
  - Un modo divertente di parlare di complessità computazionale: puzzle, matematica e algoritmi ricorsivi.
  - Algoritmi di approssimazione per problemi su grafi: Vertex Cover e Travelling Salesman Problem.
  - Protocollo di comunicazione randomizzato per stabilire la consistenza di due database.

Infine, è giusto ricordare la variegata composizione, in termini di *curriculum studiorum* e di esperienza di insegnamento, dei discenti del TFA e PAS. Gli iscritti al TFA avevano in prevalenza lauree in Informatica o Ingegneria informatica e manifestavano una buona conoscenza degli argomenti trattati. Decisamente diversa la situazione della maggioranza degli iscritti al PAS, che non avevano affrontato nei loro studi universitari insegnamenti specifici sulla teoria della computabilità. Nel caso del PAS, pertanto, come previsto dalla normativa, il nostro corso ha affiancato alla didattica della disciplina, anche una necessaria attività di verifica e consolidamento delle relative conoscenze.

### 7.3. Valutazione didattica

Non essendoci una tradizione consolidata nella formazione dei docenti di informatica in Italia, abbiamo ritenuto particolarmente importante valutare l'esito della nostra azione didattica. Attraverso l'indagine condotta abbiamo voluto soprattutto capire l'opinione, al termine del corso, dei futuri docenti in merito all'importanza e alla fattibilità di introdurre argomenti di teoria della computabilità nei programmi scolastici delle materie riferibili alla classe A042.

N.	ETICHETTA	DOMANDA
1	PRE	Quale era la tua conoscenza in ingresso sui contenuti del corso?
2	SOR	In che misura i contenuti del corso sono stati inattesi?
3	DIFF	In che misura i contenuti del corso sono stati difficili?
4	CONT	In che misura ritieni importanti i contenuti del corso?
5	METODO	In che misura ritieni efficace il metodo di insegnamento utilizzato dal docente nel corso?
6	INFLG	In che misura ritieni che il corso abbia influenzato la tua concezione generale dell'informatica?
7	LSa	In che misura ritieni che i contenuti del corso possano essere oggetto di insegnamento in un Liceo Scientifico - opzione Scienze Applicate?
8	LSb	In che misura ritieni importante insegnare i contenuti del corso in un Liceo Scientifico - opzione Scienze Applicate?
9	ITola	In che misura ritieni che i contenuti del corso possano essere oggetto di insegnamento in indirizzi o articolazioni orientati all'informatica di Istituti Tecnici?
10	ITolb	In che misura ritieni importante insegnare i contenuti del corso in indirizzi o articolazioni orientati all'informatica di Istituti Tecnici?
11	ITnola	In che misura ritieni che i contenuti del corso possano essere oggetto di insegnamento in indirizzi o articolazioni NON orientati all'informatica di Istituti Tecnici?
12	ITnolb	In che misura ritieni importante insegnare i contenuti del corso in indirizzi o articolazioni NON orientati all'informatica di Istituti Tecnici?

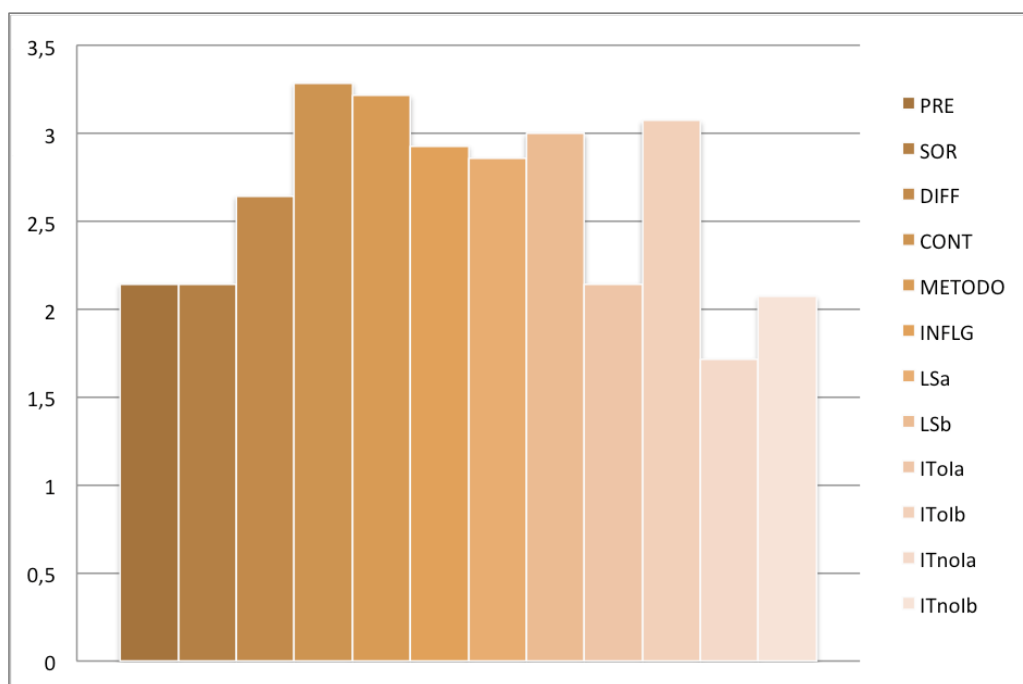
Tab. 7.1.

Una volta concluse le lezioni del corso e le relative sessioni d'esame, abbiamo chiesto ai discenti di rispondere, usando una scala di Likert a 5 punti, al questionario riportato in Tabella 7.1.

Le domande 1-2 sono volte a indagare la conoscenza in ingresso degli argomenti del corso, mentre la 3 e la 5 a valutare l'efficacia della trasmissione dei contenuti. Le domande 4 e 6, invece, servono a capire

quanto il corso abbia avuto successo nel sottolineare l'importanza culturale dei contenuti. Attraverso le domande 7-12 abbiamo chiesto l'opinione dei futuri docenti sulla fattibilità e sull'importanza di insegnare la teoria della computabilità, distinguendo le scuole in cui esistono insegnamenti di pertinenza della classe A042 in 3 categorie: liceo scientifico (opzione scienze applicate), indirizzi di istituti tecnici in cui l'informatica è tra gli insegnamenti caratterizzanti, altri indirizzi di istituti tecnici.

Hanno risposto al sondaggio, complessivamente 14 discenti, in prevalenza frequentanti del PAS. La Tabella 7.2 mostra i valori medi delle risposte.



Tab. 7.2.

Innanzitutto le risposte ci dicono che la conoscenza in ingresso della teoria della computazione supera di poco la media, e questo non ci sembra un buon viatico per l'insegnamento dell'informatica nelle scuole secondarie. D'altronde il risultato non è sorprendente, visti i titoli di studio degli aspiranti docenti di informatica che hanno svolto il PAS. In base all'esperienza fatta nella formazione degli insegnanti, riteniamo che l'elenco dei titoli di studio che consentono l'accesso alla classe A042 sia troppo ampio. I valori medio-alti delle risposte alle domande 3-6 ci confortano sul successo del corso, in modo particolare in



riferimento alla sensibilizzazione circa l'importanza dei contenuti in relazione alla disciplina. Le risposte alle domande 8, 10, 12 indicano che i discenti sono in media convinti che sia importante parlare di teoria della computazione nelle scuole, compresi gli indirizzi di istituti tecnici in cui l'informatica ha un ruolo di contorno. Questo è, a nostro avviso, il risultato migliore del corso, perché valida l'idea che il Pensiero Computazionale sia un insieme di competenze importanti per il cittadino del XXI secolo. Tuttavia, il compito viene percepito come arduo, come attestano le risposte alle domande 7, 9, 11, in particolare negli istituti tecnici. Sorprende soprattutto il risultato della domanda 9, che tendiamo a interpretare come scarsa fiducia nella propensione degli studenti degli istituti tecnici verso gli aspetti teorici di una disciplina.

Abbiamo ricavato ulteriori riscontri dalla lettura delle relazioni finali prodotte dai discenti al termine dei percorsi TFA e PAS, in cui, come previsto dalla normativa, veniva richiesto agli abilitandi di esprimere un giudizio critico su quanto appreso durante il percorso. C'è un consenso quasi unanime sul fatto che il concetto di complessità computazionale inteso come numero di passi richiesti per l'esecuzione di un algoritmo sia alla portata degli studenti di scuola superiore, e che costituisca un forte arricchimento della nozione di algoritmo. Alcuni abilitandi hanno anche provato ad ipotizzare una collocazione precisa nell'ambito di un programma di studi: terzo o quarto anno in un istituto tecnico (assumendo che nel corso del terzo anno si studino algoritmi notevoli), mentre in un liceo tratterebbero l'argomento nel corso del quinto anno, traendo vantaggio dallo studio parallelo dell'analisi matematica. Altrettanto favorevolmente vengono visti gli algoritmi elementari su grafi, sebbene vengano considerati più difficili rispetto ad altri algoritmi elementari, come quelli di ordinamento di vettori. Diverse tesi esprimono un giudizio positivo anche su algoritmi randomizzati e algoritmi di approssimazione, specie in virtù della disponibilità di esempi concreti, tratti dal materiale del corso, in problemi in cui essi possono risultare più efficaci rispetto ad algoritmi deterministici esatti. Relativamente alla teoria della calcolabilità e allo studio delle classi di complessità, le opinioni sono maggiormente dissonanti. La maggioranza dei candidati, in effetti preferisce non esporsi, definendole molto importanti ma guardandosi bene dall'inquadrarle in un possibile programma di studio. I più ottimisti si dichiarano fiduciosi

sul fatto che tali argomenti siano alla portata degli studenti dell'ultimo anno di un liceo, a loro dire più abituati alla speculazione teorica, mentre per l'istituto tecnico propongono di "selezionare un sottoinsieme degli argomenti relativi alla dicotomia P/NP". Altri abilitandi si esprimono invece in senso negativo, ritenendo le nozioni di calcolabilità e classe di complessità "troppo complesse per poter essere comprese da studenti di una scuola superiore" in quanto "comprendere davvero il non determinismo richiede delle capacità di astrazione che gli studenti di questa fascia di età non hanno". Infine, anche per i più pessimisti, la nozione dell'esistenza di problemi irrisolvibili algoritmicamente potrebbe essere data, con un taglio molto informale.

## 7.4. Conclusioni

Gli algoritmi e il complesso di teorie che ne studiano le proprietà fondamentali, sono al cuore dell'informatica. Secondo molti punti di vista epistemologici è proprio la centralità degli algoritmi nell'attività di ricerca a caratterizzare l'informatica come disciplina scientifica autonoma. Troviamo quindi paradossale il modo in cui l'idea stessa di algoritmo venga travisata da ampie porzioni del mondo della scuola, dai docenti che li identificano con i diagrammi di flusso fino agli esperti del MIUR che in un quesito della prova di accesso al TFA 2012/13 hanno ridotto la portata del concetto al puro determinismo. Il desiderio di superare questa situazione ci ha spinti a dedicare un corso agli algoritmi nei percorsi di formazione degli insegnanti, incentrato sui concetti fondamentali di calcolabilità e complessità computazionale. L'obiettivo era quello di sensibilizzare i futuri docenti sull'importanza di impartire ai loro allievi una corretta preparazione di base sugli algoritmi, ma il nostro intento è stato anche quello di fornire loro un aiuto concreto sotto forma di materiale potenzialmente adattabile all'uso nelle scuole.

Sulla base dei dati che abbiamo raccolto, tramite questionari e analisi delle relazioni finali, riteniamo di aver ottenuto risultati soddisfacenti, quantomeno rispetto all'obiettivo di dare consapevolezza della centralità agli algoritmi nell'ambito dell'informatica, sia intesa come disciplina scientifica che come materia scolastica.

## Bibliografia

- [1] ACM – CSTA, *Running on empty: The failure to Teach K-12 Computer Science in the digital age*, <http://www.acm.org/Runningonempty/>, 2013.
- [2] CALZAROSSA, M.C., CIANCARINI, P., MICH, L., SCARABOTTOLO, N., *Informatics Education in Italian High Schools*, Proc. of 5th Int. Conference in Informatics in Secondary Schools – Evolution and Perspectives (ISSEP), 31-42, 2011.
- [3] GAL-EZER, J., *A Pre-Programming Introduction to Algorithmics*, Mathematics and Computer Education, 1996, 30, 1, 61-69.
- [4] GAL-EZER, J. AND HAREL, D., *Curriculum for a high school computer science curriculum*, Computer Science Education 9(2), 114-147, 1999.
- [5] GAL-EZER, J., ZUR E., *The Concept of 'Algorithm Efficiency' in the High School CS Curriculum*, Proc. of Frontiers in Education (FIE) 2002.
- [6] GANDER W., PETIT A., BERRY G., DEMO B., ET AL., *Europe cannot afford to miss the boat*, Rep. of the joint Informatics Europe & ACM Europe Working Group on Informatics Education, <http://www.informatics-europe.org/services/reports.html>, 2013.
- [7] HROMKOVIC J., *Contributing to General Education by Teaching Informatics*, Proc. of 4th Int. Conference in Informatics in Secondary Schools – Evolution and Perspectives (ISSEP), 25-37, 2006.
- [8] MERRIT, S. M. ET AL., *ACM model of high school computer science curriculum*, Communications of the ACM 36 (5), 87-90, 1993.
- [9] TUCKER, A.B., *K-12 Computer Science: Aspirations, Realities, and Challenges*, Proc. of the 4th International Conference on Informatics in Secondary Schools - Evolution and Perspectives: Teaching Fundamentals Concepts of Informatics, 22-34, January 13-15, 2010, Zurich, Switzerland.
- [10] WING, J.M., *Computational Thinking*, Communications of the ACM 49 (3) March 2006.