

Olimpiadi Internazionali di Informatica

Syllabus per le selezioni territoriali e per le gare olimpiche

A) Premessa

Il Comitato per le Olimpiadi di Informatica ha ritenuto opportuno redigere un “syllabus” nel quale elencare in modo sintetico gli argomenti la cui padronanza è ritenuta indispensabile per gli studenti che intendono partecipare alle selezioni territoriali (regionali), alla gara olimpica nazionale e alla selezione dei componenti della squadra che parteciperà alle Olimpiadi internazionali, nonché ovviamente ai quattro “atleti” che rappresenteranno l’Italia nella gara finale.

Nella redazione del syllabus si è tenuto conto delle esperienze derivanti dalle ormai numerose partecipazioni italiane alle Olimpiadi Internazionali di Informatica, partecipazioni che hanno evidenziato come i problemi sottoposti ai concorrenti siano stati sempre focalizzati esclusivamente sugli aspetti algoritmici, a volte piuttosto sofisticati, dell’Informatica. Questa focalizzazione è presumibilmente destinata a continuare anche nei prossimi anni.

Di conseguenza il syllabus non deve essere considerato una elencazione esauriente dei concetti, dei metodi e degli strumenti dell’Informatica, ma semplicemente come un riferimento per gli studenti che intendono partecipare alle gare olimpiche con buone speranze di successo e per i docenti che li aiuteranno a raggiungere una preparazione adeguata.

Il syllabus si articola in due livelli: è preceduto da un elenco di prerequisiti e seguito da una appendice che contiene suggerimenti di vario tipo. Il primo livello è rivolto a coloro che affrontano le selezioni territoriali, mentre il secondo livello è dedicato a coloro che si cimentano nelle gare olimpiche nazionali e internazionali. Naturalmente il secondo livello include il primo. Ricordiamo comunque che il tipo di syllabus proposto è solo un elenco di strumenti concettuali ed operativi; non basta conoscerli per risolvere i problemi. E’ necessario sapere quando e come usarli, combinarli e adattarli alle situazioni contingenti. Tra le abilità più caratteristiche non esplicitamente incluse nei livelli superiori del syllabus citiamo solo le più importanti:

- 1) Comprendere/analizzare il problema per identificare gli aspetti computazionali al di là della metafora utilizzata nella descrizione
- 2) Formulare un’idea/procedura risolutiva corretta e il più possibile efficiente
- 3) Tradurre l’idea in un programma che risolva il maggior numero di casi possibili
- 4) Verificare la soluzione scrivendo degli ulteriori programmi che generano dei casi difficili da risolvere.

Infatti, secondo i regolamenti di gara, le qualità dei programmi che contano sono solo due: correttezza (fornire i risultati attesi) ed efficienza (in tempo e/o spazio).

L’esperienza insegna tuttavia che, oltre ai “trucchi”, l’adozione di qualche stile di programmazione consente in molti casi preziose economie del tempo di gara.

B) Prerequisiti

Il syllabus è preceduto da un elenco molto sintetico di argomenti la cui conoscenza è da considerare un pre-requisito per la partecipazione alle selezioni territoriali (regionali). Questi argomenti sono illustrati ampiamente nel documento dal titolo “Materiale didattico per la preparazione alla selezione scolastica (I° livello)” diffuso nell’ottobre 2007 e reperibile nel sito delle Olimpiadi all’indirizzo www.olimpiadi-informatica.it, cliccando sulla sezione “Allenamenti” e quindi su “Materiale didattico”. Il documento contiene anche numerosi esempi di quesiti e di problemi sottoposti ai partecipanti alle selezioni olimpiche negli anni scorsi, con risposte commentate. L’elenco è comunque il seguente:

Nozioni di base di aritmetica, geometria e logica

Sistemi di numerazione posizionali

Funzioni matematiche elementari (*Funzioni algebriche, esponenziali e logaritmiche*)

Nozioni di base di algebra booleana

Algoritmi (*Nozione di algoritmo, progettazione di semplici algoritmi, rappresentazione degli algoritmi mediante uno pseudo-linguaggio o mediante diagrammi*)

Memoria di un elaboratore digitale (*Rappresentazione di dati numerici, booleani, alfanumerici*)

Architettura di un elaboratore digitale (*Unità centrale, unità di input e di output, memorie di lavoro e di massa, archivi (file), reti di elaboratori*)

Linguaggi e programmi (*Linguaggio macchina e linguaggi simbolici, linguaggi ad alto livello, programmi e compilatori, software di sistema e software applicativo, sistemi operativi*)

Comprensione di programmi (*Comprensione di programmi o di porzioni di programmi scritti in un linguaggio ad alto livello, in particolare Pascal o C) che includano almeno i seguenti aspetti:*

- *Dati: tipi, valori, operazioni, costanti ed espressioni*
- *Rappresentazione degli interi in complemento a 2*
- *Variabili ed assegnamento*
- *Array (matrici) mono e multidimensionali*
- *Vita e visibilità di un identificatore*
- *Procedure e funzioni*
- *Passaggio di parametri per indirizzo e per valore*
- *Istruzioni semplici e composte*
- *Comandi di input/output elementari*
- *Tecniche per la composizione di algoritmi (sequenza, strutture condizionali, strutture iterative, ricorsione semplice e mutua).*

C) Syllabus per le selezioni territoriali (regionali)

Strutture di dati	<i>Liste, code, pile (stack), matrici (array), tabelle, alberi(binari, binari di ricerca, alberi generali), grafi</i>
Strutture di memoria	<i>Record, puntatori, strutture ad indice (array), strutture concatenate (a puntatori), gestione della memoria a heap</i>
File di testo	<i>Operazioni sui file di testo, lettura e scrittura di numeri e stringhe</i>
Implementazione di programmi	<i>Progettazione di algoritmi e redazione di programmi mediante l'uso di un linguaggio simbolico ad alto livello (Pascal, C, C++)</i>
Algoritmi di ordinamento	<i>Conoscenza di almeno uno tra i più veloci</i>
Algoritmi di ricerca su array e tabelle	<i>Ricerca sequenziale, ricerca binaria</i>
Algoritmi "divide et impera"	
Algoritmi di enumerazione (disposizioni, permutazioni)	
Algoritmi su grafi e alberi	<i>Visita di alberi</i>

D) Syllabus per le Olimpiadi italiane ed internazionali

Problemi computazionali e complessità

Tecniche di verifica dei programmi

Programmazione dinamica e tecnica greedy

Liste con applicazioni

Alberi con applicazioni

Strutture di dati per la ricerca (dizionari)

Grafi, loro rappresentazione e uso

Pile, code e visite in profondità/ampiezza di grafi

E) Appendice

1. Testi consigliati (in ordine di lettura)

- ◆ *Skiena, Revilla. Programming Challenges: The Programming Contest Training Manual, Springer-Verlag New York Inc, 2003.*
Concepito appositamente per le gare delle ACM e delle olimpiadi. Altamente consigliato. Scritto in inglese.
- ◆ *Bentley. Programming Pearls, seconda edizione, Addison Wesley, 2000.*
Realmente una perla su come programmare gli algoritmi nella pratica. Scritto in inglese.
- ◆ *Crescenzi, Gambosi, Grossi. Strutture di Dati e Algoritmi, Addison-Wesley Pearson, 2006.*
Ispirato dall'esperienza didattica sperimentata durante le olimpiadi di informatica. Pone attenzione alla programmazione e alla visualizzazione delle tecniche algoritmiche di base.
- ◆ *Sedgewick. Algoritmi in C, Addison-Wesley Pearson 2002.*
Tratta gli algoritmi e le strutture di dati tenendo presente la loro efficiente programmazione. Istruttivo perché collega la teoria alla pratica.
- ◆ *Cormen, Leiserson, Rivest, Stein. Introduzione agli Algoritmi e alle Strutture Dati, McGraw-Hill, 2005.*
Enciclopedico e impiegato in moltissime università italiane e straniere. Segue un approccio teorico e descrive gli argomenti classici dell'algoritmica. Non può mancare in biblioteca.
- ◆ *Demetrescu, Finocchi, Italiano. Algoritmi e strutture dati, McGraw-Hill, 2005.*
Testo di estrema chiarezza e sintesi. Valid alternativa al Cormen, Leiserson, Rivest, Stein.
- ◆ *Dasgupta, Papadimitriou, Vazirani. Algorithms, McGraw-Hill, 2005.*
Testo interessante per completare la propria preparazione culturale, con un punto di vista alternativo a quello dei libri precedenti. In inglese. Si può scaricare gratuitamente da <http://www.cs.berkeley.edu/~vazirani/algorithms.html>

2. Siti Web consigliati: didattica

- ◆ *Sito ufficiale degli allenamenti:* <http://allenamenti.olimpiadi-informatica.it/>
Il punto di partenza per la preparazione degli atleti italiani. Sono presentati i testi dati alle regionali e alle nazionali. Il forum associato sarà riattivato a breve e consentirà lo scambio di nozioni e conoscenze, un aspetto formativo importante per gli autodidatti.
- ◆ *Linux per le Olimpiadi di Informatica:* <http://ioi.dsi.unimi.it/linux.php>
Appunti per l'uso dell'interprete dei comandi, denominato *shell*, ossia il programma che il sistema Linux esegue non appena un utente si collega ad esso e che costituisce, per così dire, la principale interfaccia testuale tramite la quale il sistema colloquia con l'utente.
- ◆ *Introduzione alla programmazione in C:*
<http://www.science.unitn.it/~fiorella/guidac/indexc.html>
Tutorial in linea per chi vuole imparare il linguaggio C.

3. Siti Web consigliati: gare on-line e problemi

- ◆ *Internet Problem Solving Contest:* <http://ipsc.ksp.sk/>
IPSC è una gara di programmazione online per squadre costituite da tre persone al massimo. Le squadre partecipano via Internet. Esistono due distinte divisioni in cui competere: la "secondary school division" per le scuole superiori e la "open division" per tutte le età.
- ◆ *Peking University On-line Judge:* <http://acm.pku.edu.cn/JudgeOnline/>

Enorme archivio di problemi algoritmici con possibilità di sottoporre soluzioni a un sistema di valutazione in linea (on-line judge). Molto presenti anche testi delle trascorse edizioni di IOI e CEOI. Ottimo per allenarsi!

- ◆ *Programming Contest Problems Archive*: <http://www.inf.bme.hu/contests/tasks/>
Raccolta di riferimenti ai problemi dati nelle varie gare regionali e nazionali di vari paesi, nonché alle gare internazionali. Seguendo i vari riferimenti, è disponibile un'enorme quantità di problemi con cui allenarsi.
- ◆ *Top Coder*: <http://www.topcoder.com/>
E' praticamente una vera e propria arena di programmazione, in cui si incontrano i migliori programmatori on-line ed ex olimpici. Le gare sono organizzate mensilmente con un sistema di classifica dei migliori, i quali hanno la possibilità di vincere premi. Consigliata la sezione "algorithm & high school competition".
- ◆ *USACO Training Program Gateway*: <http://train.usaco.org/>
Occorre registrarsi per accedere ai vari livelli di gara. Esiste la possibilità di allenarsi con esercizi di livello crescente: per passare al livello successivo occorre aver risolto gli esercizi del livello corrente. Le gare sono usate per selezionare la squadra americana.

4. Software : suggerimenti

Preferibilmente usare il linguaggio C/C++ (l'uso del Pascal non è tuttavia discriminante nella gara nazionale)

Si consiglia caldamente l'adozione di Linux, preferibilmente utilizzando la distribuzione Ubuntu Linux

Distribuzione Linux Ubuntu: <http://www.ubuntu.com/>

Si trova tutto quanto necessario per gareggiare e allenarsi: compilatore, debugger, editor.

Per chi preferisce Windows, l'ambiente di sviluppo (lo stesso delle selezioni territoriali) è reperibile su :

<http://81.208.32.83:8080/oii/2008/selezioneRegionale/softwareRegionali.htm>

Si trova tutto quanto necessario per gareggiare e allenarsi: compilatore, debugger, editor.