

Syllabus

Olimpiadi di Informatica a Squadre

Versione 2 -- 19 maggio 2019

Livello 1.

- Tipi di dato primitivo (e.g. int, char, double, bool)
- Array mono-dimensionali (e.g. int[], char[], double[], bool[])
- Branching (costrutti if/else)
- Cicli limitati (costrutti for semplici)

Livello 2.

- Array multi-dimensionali (e.g. int[][])
- Strutture dati coda e pila
- Cicli illimitati (costrutto while)
- Funzioni e ricorsione esaustiva (e.g. elencare le permutazioni)
- Ricerca binaria
- Algoritmi di ordinamento quadratici (e.g. bubble sort)
- Stringhe e ricerca quadratica di una sottostringa in una stringa
- Algoritmo di Euclide per il massimo comun divisore
- Concetti base di geometria e aritmetica

Livello 3.

- Contenitori standard (e.g. vector, set, map)
- Divide et impera
- Programmazione dinamica su array mono- o multi-dimensionali
- Visite di grafi (BFS, DFS)
- Algoritmi di ordinamento efficienti (sort o qsort)
- Ottimizzazione approssimata tramite tecniche euristiche

Livello 4.

- Struttura dati union-find per insiemi disgiunti
- Strutture dati per query su range
- Minimo antenato comune su alberi
- Algoritmi su grafi: cammini minimi, albero ricoprente
- Programmazione dinamica su grafi aciclici
- Backtracking (algoritmi branch and bound)

Livello 5.

- Tutti gli argomenti non menzionati nei livelli precedenti

Syllabus

Olimpiadi di Informatica a Squadre

Version 2 --- May 19th, 2019

Level 1.

- Primitive data types (e.g. int, char, double, bool)
- One-dimensional arrays (e.g. int[], char[], double[], bool[])
- Branching (if/else statement)
- Bounded loops (simple for statement)

Level 2.

- Multi-dimensional arrays (e.g. int[][])
- Queue and stack data structures
- Unbounded loops (while statement)
- Functions and exhaustive recursion (e.g. listing permutations)
- Binary search
- Quadratic sorting algorithms (e.g. bubble sort)
- Strings and quadratic substring search
- Euclidean algorithm for the greatest common divisor
- Basic concepts of geometry and arithmetic

Level 3.

- Standard containers (e.g. vector, set, map)
- Divide et impera
- Dynamic programming on one- or multi-dimensional arrays
- Graph traversals (BFS, DFS)
- Efficient sorting algorithms (sort or qsort)
- Approximate optimization through heuristics techniques

Level 4.

- Disjoint-set data structure (union-find)
- Data structures for range queries
- Least common ancestor in trees
- Graph algorithms: shortest path, spanning tree
- Dynamic programming on acyclic graphs
- Backtracking (branch and bound algorithms)

Level 5.

- All other topics not mentioned in previous levels