

# Argomenti principali del corso

- Utilizzo di strumenti avanzati del C++ (template, ereditarietà, const/static, ecc...)
- Implementazione di dinamiche che tengono conto della complessità degli algoritmi (ordinamento, ricorsione, divide et impera)
- Implementazione ed utilizzo di strutture dati dinamiche mediante algoritmi specifici (pile, code, BST, Grafi, ecc...)

# Esempi di domande

- Qual è la complessità del seguente algoritmo?
- Qual è l'ordine di grandezza della funzione  $G = T1 * T2$  ?
- Quale riga genera un errore di compilazione?
- Qual è l'output del seguente codice ?
- Considerando il seguente BST. Indicare quale sarebbe il padre del nodo X dopo la rimozione del nodo Y....
- Quando si dice che una classe è astratta ?
- Cosa intendiamo dire quando diciamo che un algoritmo X è asintoticamente più efficiente di un altro algoritmo Y ?

# Prova pratica: come mi esercito?

- Esercizi svolti in aula, e possibili varianti
- Esercizi elencati alla fine di alcuni blocchi di slides
- Implementazioni suggerite nelle slides
- Esercizi proposti nei libri di testo (Aguilar/Cormen)

# Due esempi dal libro di testo

- Definire una funzione membro della classe *albero* per visualizzare tutti i nodi di un albero binario di ricerca il cui valore del campo chiave sia maggiore di un valore accettato in input.
- Definire una classe *albero* che restituisca una istanza della classe *albero* che rappresenta il simmetrico dell'oggetto *albero*. Per simmetrico intendiamo un albero in cui ogni nodo ha sottoalberi sinistro e destro invertiti.