

Tempo concesso per lo svolgimento di una sola parte: 1h 30min.

Tempo concesso per lo svolgimento dell'esame completo: 3h.

Lo pseudocodice non commentato e le risposte non motivate non saranno prese in considerazione.

Parte 1

Esercizio 1

Si definisca la notazione " Ω " e se ne discuta il significato.

Si risponda poi ad ognuna delle seguenti domande, motivando la risposta:

1. Se un algoritmo ha tempo di esecuzione $\Theta(n^2)$ nel caso peggiore, posso dedurre che nel caso migliore terminerà in $\Theta(n^2)$ passi?
2. Se si dimostra che un algoritmo ha tempo di esecuzione $\Omega(n^2)$ nel caso migliore, è possibile che *in qualche caso* l'algoritmo termini in $O(n)$ passi?
3. Se si dimostra che un algoritmo ha tempo di esecuzione $\Omega(n^2)$ nel caso peggiore, è possibile che *in qualche caso* l'algoritmo termini in $O(n)$ passi?

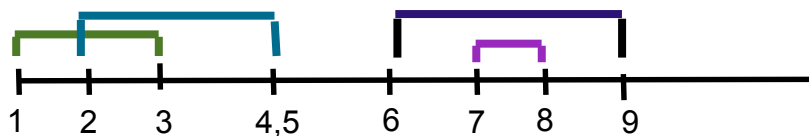
Esercizio 2

Si determini il tempo di esecuzione $T(n)$ della seguente funzione:

```
analizzami(int n)
  c=m=1
  for i=1 to n do
    m=3*m
    for j=1 to m do
      c++
    endfor
  endfor
```

Esercizio 3

Dati n intervalli della retta reale, si descriva un algoritmo che calcoli la lunghezza della loro unione in $O(n \log n)$ passi. Per esempio, la lunghezza dell'unione dei quattro intervalli $[1, 3]$, $[2, 4.5]$, $[6, 9]$, e $[7, 8]$ è 6.5.



Si delinei l'idea algoritmica, e si individui un'invariante di ciclo per risolvere il problema con un algoritmo corretto. Si dimostri che l'algoritmo è corretto, usando l'invariante introdotto per verificare il ciclo, e si analizzi il tempo di esecuzione.

Suggerimento: utilizzare un array di coppie $\langle \text{inizio}, \text{fine} \rangle$ come struttura dati per memorizzare gli intervalli.

Introduzione agli Algoritmi
Appello del 12 Giugno 2012
Prof. Emanuela Fachini ed Irene Finocchi

Tempo concesso per lo svolgimento di una sola parte: 1h 30min.

Tempo concesso per lo svolgimento dell'esame completo: 3h.

Lo pseudocodice non commentato e le risposte non motivate non saranno prese in considerazione.

Parte 2

Esercizio 4

Dimostrare che ogni algoritmo di ricerca di un elemento in un vettore ordinato, nel modello basato su confronti, esegue nel caso peggiore $\Omega(\log n)$ confronti.

Esercizio 5

Supponete di memorizzare n valori in un albero binario di ricerca oppure in un albero rosso-nero.

Compilate la tabella sottostante, specificando il tempo di esecuzione asintotico per ciascuna operazione. Commentate brevemente i risultati inseriti nella tabella, facendo riferimento al caso peggiore.

Operazione	ABR	Rosso-nero
<i>Trovare un elemento</i>		
<i>Costruire la struttura</i>		
<i>Cancellare un elemento</i>		
<i>Trovare il minimo</i>		

Esercizio 6

Si consideri un ABR T rappresentato tramite puntatori.

Si progetti un algoritmo Left-Chain che trasformi T in una catena sinistra, cioè in un ABR in cui ogni elemento, tranne la radice, è figlio sinistro. Si delinei l'idea algoritmica prima di scrivere lo pseudocodice. Si analizzino inoltre *correttezza* e *tempo di esecuzione* dell'algoritmo progettato.

Suggerimento: utilizzare una funzione Left-Rotate(T,p), che prende in input il puntatore alla radice e un puntatore p a un nodo di T ed esegue una rotazione a sinistra usando p come perno. La funzione ha come preconditione che il figlio destro di p sia non nullo.