

concesso. In 50 min. Le risposte non motivate non saranno prese in considerazione.

Parte 1

Esercizio 1

1. Si confronti $n \log n$ con n^2 : $n \log n = O(n^2)$ o $n^2 = O(n \log n)$?
2. Si confronti $n \log n$ con n : $n \log n = O(n)$ o $n = O(n \log n)$?
3. E vero che $n \log n = O(n \log^2 n)$?
4. $n^2 + n \log n = O(\dots)$
5. $n + n \log n = O(\dots)$
6. $n^2 + \log \log n = O(\dots)$
7. Trova una funzione $g(n)$ such that $n^2 = \Omega(g(n))$
8. Ordina le seguenti funzioni in ordine di tasso di crescita: n^2 , 2^n , $n \log n$, $n!$, $n^6 + n^2$, n^4 , 4^n , n , n^n .

Esercizio 2

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

```
fun(int n)
  i=1, a = 1;
  while(i <= n) {
    j = 10;
    while(j <= 1000) {
      a = 2a;
      j++;
    }
    i++;
  }
```

Esercizio 3

Sia S un insieme di n valori reali distinti. Si assuma che gli elementi di S siano memorizzati in un array A "*k-disordinato*": tutti gli elementi in A sono ordinati, in ordine crescente, tranne i k elementi più grandi di S che possono occupare posizioni arbitrarie in A . Assumendo che k sia costante, progettare ed analizzare un algoritmo efficiente per ordinare A .

considerazione.

Parte 2

Esercizio 4

Per ognuno dei compiti sotto elencati fornite la complessità asintotica nel caso peggiore del miglior algoritmo che lo svolge, usando la notazione O . Si fornisca una breve giustificazione del risultato, indicando l'implementazione scelta per la struttura dati.

1. Inserire un elemento in una coda di priorità (MaxHeap).
2. Estrarre il massimo in una coda di priorità (MaxHeap).
3. Inserire una chiave in un dizionario.
4. Cercare una chiave in un dizionario.
5. Ordinare gli elementi di un insieme.

Esercizio 5

Dimostrare che ogni algoritmo per l'ordinamento di n elementi basato su confronti ha tempo di esecuzione $\Omega(n \log n)$ nel caso peggiore.

Esercizio 6

Primo canale

Sia T un albero rosso-nero contenente n valori. Progettare un algoritmo che calcoli il numero minimo di nodi rossi su un cammino radice-foglia. Si analizzino correttezza e complessità dell'algoritmo proposto.

Secondo canale

Sia T un albero AVL contenente n valori. Progettare un algoritmo che calcoli il numero minimo di nodi con fattore di bilanciamento $+1$ su un cammino radice-foglia. Si analizzino correttezza e complessità dell'algoritmo proposto.