

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

Chi ha superato la prova intermedia (esonero) deve svolgere solo gli esercizi della Parte 2 (tempo concesso: 1h 30m). Gli altri studenti possono svolgere solo gli esercizi della Parte 1 (tempo concesso: 1h 30m), completando l'esame in luglio, oppure l'esame completo (tempo concesso: 3h).

Le risposte non motivate non saranno prese in considerazione. Negli esercizi di progettazione, prima di passare allo pseudocodice descrivete l'idea algoritmica sottostante. Per tutti gli algoritmi progettati è necessario analizzare correttezza e tempo di esecuzione.

Parte 1

Esercizio 1

Definire la notazione Θ e discuterne il significato.

Rispondere poi ad ognuna delle seguenti domande, motivando la risposta:

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

Esercizio 2

- a. Preso un vettore A di n valori reali, progettare un algoritmo che restituisca, se esistono, gli indici di due elementi in A la cui somma è zero. L'algoritmo deve terminare in $O(n \log n)$ passi nel caso peggiore.
- b. Delineare l'idea algoritmica per risolvere il caso in cui si cercano tre elementi a somma zero. Il tempo di esecuzione dovrebbe essere $O(n^2 \log n)$.

Esercizio 3

Considerare le seguenti funzioni:

<pre>fun1(int k) { if k<=1 then return; fun1(k/9); fun1(k/9); fun1(k/9); }</pre>	<pre>fun2(int n) { i=n while (i>1) do { fun1(i) i = i/2 } }</pre>
---	--

1. Qual è il tempo di esecuzione di fun1 in funzione di k ?
 2. Qual è il tempo di esecuzione di fun2 in funzione di n ?
-

Parte 2

Esercizio 4

Dimostrare che ogni algoritmo di ordinamento basato su confronti esegue nel caso peggiore $\Omega(n \log n)$ confronti.

Esercizio 5

Supponete di memorizzare n valori in un max-heap oppure in un array ordinato.

Compilate la tabella sottostante, specificando il tempo di esecuzione asintotico nel caso peggiore per ciascuna operazione. Stabilite, di conseguenza, quale delle due strutture dati è preferibile per ognuna delle operazioni considerate oppure se la scelta è indifferente.

operazione	tempo di esecuzione max-heap	tempo di esecuzione array ordinato
<i>trovare il massimo</i>		
<i>cancellare un elemento</i>		
<i>costruire la struttura</i>		
<i>trovare il minimo</i>		

Esercizio 6

- A. Dati un ABR T , rappresentato tramite puntatori ai figli, ed un valore r , progettare un algoritmo che determini il numero di chiavi di T minori di r .
- B. Come modifichereste l'algoritmo per avere invece il numero di chiavi maggiori di un dato valore s ?
- C. Delineare l'idea algoritmica nel caso si volesse determinare il numero di valori nell'intervallo $[r,s]$.