Introduzione agli Algoritmi
Appello esame del 2 luglio 2015
Prof. Emanuela Fachini (canale 1) e Prof. Irene Finocchi (canale 2)
Parte 1

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

Esercizio 1

Per ogni affermazione, si scriva se è vera o falsa. La verità di un'affermazione va provata, basandosi sulle definizioni di O, Θ e Ω. La falsità va dimostrata con un contro esempio, cioè fornendo le funzioni che falsificano l'affermazione.

1. $f(n) = \Theta(n)$ e $g(n) = \Omega(n) \Rightarrow f(n)g(n) = \Omega(n^2)$
2. $f(n) = \Omega(n)$ e $g(n) = O(n^2) \Rightarrow f(n)/g(n) = O(n)$
3. $f(n) = O(\log n) \Rightarrow 2^{f(n)} = O(n)$

Esercizio 2

Si consideri la seguente funzione:

```
fun (array A, int i, int f) {
        n = f-i+1
        t=n;
        while t≥1 do t = t-2;
        if (n < 1) then return 1;
        else return i + 2*fun (A, i, (i+f)/2);
}
```

Qual è il tempo di esecuzione di *fun* in funzione di n? Si imposti e si risolva la relazione di ricorrenza.

Esercizio 3

Dato un array A di interi si scriva un algoritmo che li riposiziona nell'array, operando **in loco,** in modo tale che i pari siano elementi di indice pari e i dispari siano di indice dispari. Se ci sono più pari che dispari o viceversa questi vanno accumulati alla fine dell'array A.
Per esempio se l'array, con indici che partono da 1, inizialmente è:
A = [50, 47, 92, 78, 76, 7, 60, 36, 59, 30, 50, 43]
il risultato potrebbe essere questo
A = [47, 50, 7, 78, 59, 92, 43, 36, 76, 30, 50, 60]
Si scriva un algoritmo AltPariDisp(A) che in O(n) passi realizza il riposizionamento voluto.

Introduzione agli Algoritmi
Prova intermedia 14 Aprile 2015
Prof. Emanuela Fachini (canale 1) e Prof. Irene Finocchi (canale 2)
Parte II

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

Esercizio 4

Si dimostri che un AVL con n nodi ha altezza O(lg n). La dimostrazione deve essere completa in tutti i suoi passaggi ed autocontenuta.

Esercizio 5

Un insieme di chiavi è memorizzata in un Max-heap e in un ABR T.
Quale delle due strutture dati è più efficiente se vogliamo stampare le chiavi in ordine decrescente?
Si scrivano i due algoritmi e li si analizzino dal punto di vista del tempo di esecuzione asintotico.

Esercizio 6

Si scriva un algoritmo TerTreeSearch(T,k) che prende in input la radice di un albero ternario T e una chiave k, restituendo il nodo che ha la chiave k o NIL se la chiave non è presente. Non ci sono chiavi duplicate in T.
Un albero ternario è concettualmente identico ad un albero binario ma ogni nodo x ha al più tre figli x.left, x.right e x.center e ha due chiavi x.key1 e x.key2, con x.key1 < x.key2. I nodi nel sotto albero sinistro di x contengono chiavi k1 e k2 tali che k1 < k2 < x.key1, i nodi nel sotto albero destro k1 e k2 tali che x.key2 < k1 < k2 e quelle nel sotto albero centrale le chiavi k1 e k2 tali che x.key1 < k1 < k2 < x.key2.
Qual'è la complessità di tempo asintotica dell'operazione di ricerca?

Per esempio, la ricerca della chiave 61 deve restituire il puntatore al nodo che ha le chiavi 61,65