

Cognome e Nome:

Numero di Matricola:

Spazio riservato alla correzione

1	2	3	4	5	6	7	totale
/18	/10	/12	/18	/15	/15	/12	/100

1. Relazioni asintotiche

a) Si dimostri che $3n^5 + 10n^3 + 3 = O(n^5)$ ricavando le costanti c ed n_0 .

b) Si dimostri che se $f(n)=O(g(n))$ e $h(n)=O(g(n))$ allora $f(n)+h(n)=O(g(n))$

- c) Si consideri la relazione di ricorrenza: $T(n) \leq c_0$ se $n \leq n_0$, $T(n) \leq 4 T(n/3) + cn$, altrimenti (c e c_0 sono costanti). Si fornisca una funzione $h(n)$ tale $T(n) = O(h(n))$. Giustificare la risposta.

2. Si analizzi il tempo di esecuzione nel caso pessimo dei seguenti segmenti di codice fornendo una stima asintotica per esso. Si giustifichi la risposta.

a.

```
for (i=1; i<n; i=i*2)
  stampa i;
```

```
b. for(i=0;i<n;i=i+1){
    j=0;
    while(j<n/2){
        stampa i;
        stampa j;
        i=i+1;
        j=j+1;
    }
}
```

3. Si consideri la seguente funzione che prende in input un array di numeri e tre interi non negativi. Si descriva cosa fa la funzione e analizzarne la complessità.

```
f( a,inizio, arbitrario, fine ):
if(arbitrario != fine) {
    t=a[arbitrario];
    a[ arbitrario]=a[fine];
    a[fine]=t;
}
i=inizio;
j=fine-1;
while(i <= j) {
while((i <= j) && (a[i] <= a[fine]))
    i=i+1;
while((i <= j) && (a[j] >= a[fine]))
    j=j-1;
if (i < j){
    t=a[i];
    a[i]=a[j];
    a[j]=t;
}
}
if(i != fine){
    t=a[fine];
    a[fine]=a[i];
    a[i]=t;
}
```

```
return i;
```

4. Divide et impera:

Dato un array \mathbf{A} di n interi si dice che \mathbf{A} contiene un *salto* se esiste un indice i , $0 \leq i < n-1$, tale che $\mathbf{A}[i] < \mathbf{A}[i+1]$. Si noti che se $\mathbf{A}[0] < \mathbf{A}[n-1]$ allora l'array contiene certamente un salto altrimenti l'array sarebbe ordinato in ordine non crescente e risulterebbe $\mathbf{A}[0] \geq \mathbf{A}[n-1]$.

Si descriva un algoritmo che, dato in input un array \mathbf{A} tale che $\mathbf{A}[0] < \mathbf{A}[n-1]$, trovi un salto in \mathbf{A} , ovvero restituisca un indice i , $0 \leq i < n-1$, tale che $\mathbf{A}[i] < \mathbf{A}[i+1]$. L'algoritmo deve avere complessità $O(\log n)$.

5. Si scriva lo pseudocodice di un algoritmo ricorsivo che, dato un albero binario, restituisca l'altezza della radice.

Introduzione agli Algoritmi ed alle Strutture Dati
Anno Accademico 2014/2015
Appello del 12/6/2015

6. Heap:

a) Si definisca in modo chiaro e conciso cosa è un heap e si dica cosa è un heap implicito.

b) Si descrivano i passi eseguiti dall'algoritmo enqueue dell'heap. Si dica qual è il tempo di esecuzione dell'algoritmo di enqueue.

7. Si provi che *ogni* algoritmo di ricerca per confronti su un array ordinato richiede $\Omega(\log n)$ confronti.