

**008AA – ALGORITMICA E LABORATORIO**  
Primo Compitino 4 Aprile 2018

Cognome Nome:

N. Matricola:

Corso: A B

**Esercizio 1.** (6 punti) Si progetti un algoritmo basato sulla tecnica *Divide et Impera* il cui costo in tempo sia definito dalla ricorrenza:

- $T(n) = 2T(n/2) + \Theta(\log^2 n)$ , se  $n > 1$
- $T(n) = \Theta(1)$  altrimenti.

e si risolva la ricorrenza.

**Esercizio 2.** (6 punti) Si progetti un algoritmo che dati due array ordinati  $A$  e  $B$  rispettivamente di  $n$  e  $m$  elementi interi, conti il numero di elementi condivisi dai due array.

**Esercizio 3.** (6 punti) Si consideri l'algoritmo seguente il cui calcolo è avviato con la chiamata iniziale `Quick_SortB(A, 1, n)`:

```
Quick_SortB(A, p, r)
if (p < r) {
    k = [(p + r)/2];
    s = Quick_Select (A, p, r, k);
    q = PartitionB (A, p, s, r);
    Quick_SortB (A, p, q-1);
    Quick_SortB (A, q+1, r);
}
```

ove la funzione `PartitionB` è uguale a `Randomized_Partition` del `QuickSort` visto in classe, eccetto che il pivot utilizzato è quello in posizione  $s$ . Si commenti il funzionamento di `Quick_SortB` e si ricavi l'equazione di ricorrenza associata. Si risolva inoltre tale equazione nel caso medio e nel caso pessimo.

**Esercizio 4.** (6 punti) Sia dato un array di interi  $A[5, 13, 1, 8, 6, 12]$  si applichi ripetutamente l'algoritmo *Max\_Heap\_Insert* per la costruzione di un Heap di massimo, mostrando la configurazione del vettore dopo ogni chiamata.

**Esercizio 5.** (6 punti) Si consideri il problema che dato un array  $A[1, n]$ , non ordinato, di interi positivi e negativi, determini se esistono due indici  $i$  e  $j$ ,  $i < j$ , tali che  $A[i] + A[j] = 0$  e, nel caso, li restituisca in output; altrimenti, restituisca la coppia  $(-1, -1)$ . Si stabilisca un limite inferiore al problema usando la tecnica dell'*albero di decisione*. Infine, si progetti un algoritmo efficiente, si valuti la sua complessità asintotica in tempo, e si valuti la significatività del limite inferiore determinato.