

008AA – ALGORITMICA E LABORATORIO

Primo Appello 22 giugno 2018

Cognome Nome:

N. Matricola:

Corso: A B

**Esercizio 1.** ( $4+3+3$  punti) Si progetti un algoritmo ricorsivo basato sulla tecnica *Divide et Impera* che calcola il secondo elemento più grande di un vettore  $A$  di  $n$  interi, senza modificarlo.

Si scriva la relazione di ricorrenza che descrive il suo costo in tempo al caso peggior, e la si risolva.

**Esercizio 2.** ( $2+3+3$  punti) Data la seguente procedura ricorsiva, definita in funzione di un parametro (globale) intero  $a > 0$ :

```
foo( A, n )
{
  if ( n <= 1 ) return 5;
  InsertionSort(A,n);
  for i = 1 to a
    foo(A,n/2);
}
```

1. Scrivere la relazione di ricorrenza che descrive la complessità in tempo al caso peggior di  $\text{foo}(A,n)$  in funzione di  $a$ .
2. Risolvere la relazione di ricorrenza al variare di  $a$ .
3. Stabilire per quali valori di  $a$  la procedura ha complessità in tempo al caso peggior  $O(n^3)$ .

**Esercizio 3.** ( $6$  punti) Sia dato il grafo diretto aciclico  $G(V, E)$  costituito da 7 vertici numerati da 1 a 7 e i seguenti archi  $E = \{(1, 2), (1, 3), (1, 4), (2, 3), (4, 3), (3, 5), (6, 7), (7, 2)\}$ . Si calcoli l'ordinamento topologico di  $G$  simulando il funzionamento dell'algoritmo `TOPOLOGICAL_SORT`, e assumendo che le liste di adiacenza siano ordinate per vertice destinazione.

**Esercizio 4.** ( $3+3$  punti)

1. Si dia la definizione di riduzione polinomiale tra due problemi decisionali  $P$  e  $Q$ .
2. Si scriva lo pseudo-codice dell'algoritmo `Build_heap` (assumendo di avere a disposizione l'algoritmo `Heapify`) per la costruzione di un heap di massimo su  $n$  interi, e si dimostri che la complessità in tempo è  $O(n)$ .