

008AA – ALGORITMICA E LABORATORIO-B
Prima Prova di Verifica Intermedia - 1 aprile 2019

Cognome Nome:

N. Matricola:

Esercizio 1 (punti 6)

[La soluzione corretta di questo esercizio è requisito per la correzione del resto del compito]

Si scrivano lo pseudocodice di MergeSort (omettendo quello di Merge) e la relazione di ricorrenza che ne descrive la complessità in tempo, motivando la risposta. Si risolva infine la suddetta relazione di ricorrenza.

Esercizio 2 (punti 6)

Si scriva lo pseudocodice di un algoritmo *Divide et Impera* efficiente che, dato array A di interi $A[1], A[2], \dots, A[n]$, restituisca il numero di indici i per cui $i \leq A[i]$.

Si calcoli la complessità dell'algoritmo proposto.

Si calcoli il limite inferiore al problema utilizzando l'albero di decisione. Si dica se il limite inferiore ottenuto è significativo e, in caso negativo, si fornisca un limite inferiore migliore, o un algoritmo migliore.

Esercizio 3 (punti 6)

Si mostrino i passaggi dell'esecuzione dell'Algoritmo BuildMaxHeap sul vettore di interi $A=[10,2,5,13,18,9,20,1,40]$. In particolare si mostri (i) per ogni iterazione, l'albero e i nodi a vario titolo coinvolti, e (ii) il vettore A risultante al termine dell'esecuzione.

Esercizio 4 (punti 6)

Si scriva la relazione di ricorrenza che descrive la complessità in tempo del seguente algoritmo la cui chiamata iniziale è $\text{Ri-Sort}(A, 1, n)$. Si confrontino poi, motivando la risposta, le complessità asintotiche di $\text{Ri-Sort}(A, n)$ e di $\text{SelectionSort}(A, n)$ al variare dell'intero K .

```
Ri-Sort(A,p,r)
if (p < r) then
  {
    q=(p+r)/2;
    for i=1 to K do {
      Ri-Sort(A,p,q);
      Ri-Sort(A,q+1,r);
    }
    MERGE(A,p,q,r);
  }
```

Esercizio 5 (punti 3)

Un ABCB (Albero Binario Completamente Bilanciato) è un albero binario in cui tutti i nodi, eccetto le foglie, hanno esattamente due figli, e le foglie si trovano tutte allo stesso livello. Si dimostri per induzione su h che un ABCB di altezza h ha 2^h foglie.

Esercizio 6* (punti 3)

[Si consiglia di risolvere questo esercizio per ultimo]

Sia s_i l' i -esimo elemento più piccolo di un array A , e sia s_j il j -esimo elemento più piccolo (ovvero, s_i e s_j sono gli elementi che si troveranno in posizione i e j nell'array ordinato).

Si scriva, motivando la risposta, la probabilità che s_i e s_j si confrontino tra loro in un'esecuzione di QuickSort.