008AA – ALGORITMICA E LABORATORIO-A
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 QuickSort (omettendo quello di Partiziona) e la relazione di ricorrenza che ne descrive la complessità in tempo al caso ottimo, motivando la risposta. Si risolva infine la suddetta relazione di ricorrenza.

Esercizio 2 *(punti 6)*

Un array A di n interi distinti si definisce **ciclicamente ordinato** se esiste un indice i, $1 \leq i \leq n$, tale che la sequenza $a[i], a[i+1], \ldots, a[n], a[1], \ldots, a[i-1]$ è ordinata in modo crescente. Ad esempio, l'array $A =$ [12, 14, 20, 1, 3, 7, 10, 11] è ciclicamente ordinato per $i = 4$. Si consideri il problema di trovare la posizione i.

Scrivere lo pseudocodice di un algoritmo efficiente di tipo *Divide et impera* per il problema precedente. Calcolare la complessità al caso pessimo dell'algoritmo proposto indicando, e risolvendo, la corrispondente relazione di ricorrenza.

Esercizio 3 *(punti 6)*

Si mostrino i passaggi dell'esecuzione dell'Algoritmo BuildMaxHeap sul vettore di interi A=[13,1,40,18,2,10,9,20,5]. 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 `Boh-Sort(A,1,n)`, e K un intero noto. Si confrontino poi, motivando la risposta, le complessità asintotiche di `Boh-Sort(A,n)` e di `SelectionSort(A,n)` al variare dell'intero `K`.

```
Boh-Sort(A,p,r)
if (p < r) then
    {
    q=(p+r)/2;
    Boh-Sort(A,p,q);
    if (q-p+1) < K
        SelectionSort(A,p,q);
    else
        Boh-Sort(A,p,q);
    MERGE(A,p,q,r);
    }
```

Esercizio 5 *(punti 3)*

Un ABCB (Albero Binario Completamente Bilanciato o Completo) 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 d che un ABCB di profondità d ha $2^{d+1} - 1$ nodi.

Esercizio 6* *(punti 3)*
[Si consiglia di risolvere questo esercizio per ultimo.]

Sia

$$g(n) = \sum_{j=0}^{\log_b n - 1} a^j f(n/b^j),$$

con a e b costanti positive. Si dimostri che, se $f(n) = O(n^{\log_b a - \epsilon})$ per qualche costante $\epsilon > 0$, allora $g(n) = O(n^{\log_b a})$.