

**Introduzione agli Algoritmi**  
**Prova del 30 gennaio 2018**  
**Proff. E. Fachini e R. Petreschi**  
**I PARTE**

**Le soluzioni degli esercizi scritte in modo illeggibile o in cui compaiano solo conti o pseudocodice senza commenti e risposte non motivate saranno valutati 0. Prima di descrivere un algoritmo in pseudocodice si deve delineare l'idea algoritmica. Inoltre deve essere precisato l'input e l'output atteso da eventuali singole funzioni utilizzate, oltre agli eventuali vincoli sul loro input (precondizioni). Infine si deve calcolare il tempo asintotico di esecuzione di ogni algoritmo descritto.**

1. Analizzare, nel caso in cui tutti gli elementi nell'array di input siano uguali, il tempo di esecuzione dei seguenti algoritmi di ordinamento:
  - a. Mergesort
  - b. Bubblesort
  - c. Insertionsort
  - d. Heapsort
  
2. Si determinino i termini dominanti nelle seguenti espressioni, che possiamo supporre derivanti dall'analisi di algoritmi per problemi su  $n$  elementi, e li si utilizzino per esprimere un limite superiore alla crescita asintotica dell'espressione stessa:

espressione	termine dominante	$O(\dots)$
$500n + 100n^{1,5} + 50n\log_{10}n$		
$n^2\log_2n + n(\log_2 n)^2$		
$n\log_3n + n\log_2 n$		
$100n\log_3n + n^3 + 100n$		

3. Un array  $A$  di dimensione  $n$  è quasi ordinato se l'  $i$ -esimo numero più piccolo, con  $1 < i < n$ , occupa la posizione  $i - 1$ ,  $i$ , oppure  $i + 1$ . Il minimo occupa la posizione 1 o 2, e il massimo la posizione  $n-1$  o  $n$ . Dato un array quasi ordinato, progettare un algoritmo, il più possibile efficiente, per ordinare l'array, ed analizzarne correttezza e tempo di esecuzione.

**Introduzione agli Algoritmi**  
**Prova del 30 gennaio 2018**  
**Proff. E. Fachini e R. Petreschi**

**II PARTE**

**Le soluzioni degli esercizi scritte in modo illeggibile o in cui compaiano solo conti o pseudocodice senza commenti e risposte non motivate saranno valutati 0. Prima di descrivere un algoritmo in pseudocodice si deve delineare l'idea algoritmica. Inoltre deve essere precisato l'input e l'output atteso da eventuali singole funzioni utilizzate, oltre agli eventuali vincoli sul loro input (precondizioni). Infine si deve calcolare il tempo asintotico di esecuzione di ogni algoritmo descritto.**

1. Presentare l'algoritmo MERGESORT in forma ricorsiva, spiegarne la correttezza e dettagliarne il tempo di esecuzione.

2. Si determini il tempo di esecuzione asintotico  $T(n)$  della seguente funzione:

```
analizzami (A, i, j)
if (i < j) then
    ne = j-i+1
    Analizzami (A, i, i+ne/4)
    Analizzami (A, j-ne/4,j)
```

3. Si scriva la relazione di ricorrenza che esprime il tempo di esecuzione asintotico e la si risolva con il metodo della sostituzione.

4. Date due sequenze binarie  $X = x_1, x_2, \dots, x_n$  e  $Y = y_1, y_2, \dots, y_n$  diciamo che  $X$  precede in ordine quasi lessicografico  $Y$ ,  $X < Y$ , se esiste un indice  $h$ ,  $1 \leq h \leq n$  tale che  $x_h < y_h$  con  $x_i = y_i$  per ogni  $1 \leq i \leq h-1$ . Per esempio  $0010 < 0011$  e  $0010 < 1011$ . Si dia una regola per etichettare con sequenze binarie i vertici di un albero binario in modo tale che la sequenza delle etichette lette in preordine rispetti l'ordinamento lessicografico.