

APPELLO DEL 22 GENNAIO 2009

GLI STUDENTI ESONERATI DEVONO SVOLGERE SOLO GLI ESERCIZI 3 E 4. TEMPO CONCESSO: 1H 45 MIN.
 CHI SOSTIENE TUTTO L'ESAME DEVE SVOLGERE TUTTI GLI ESERCIZI. TEMPO CONCESSO: 3H.

Esercizio 1. Progettare un algoritmo efficiente che, dati un intero k e un array V di n interi positivi ordinati in modo non decrescente, calcoli il numero di coppie di elementi di V la cui somma è maggiore di k . Calcolare il tempo di esecuzione dell'algoritmo proposto ed argomentare la sua efficienza.

Esercizio 2. Siano T ed S due array tali che $|T| = n$ e $|S| = \lfloor \log_2 |T| \rfloor$. Considerare le seguenti procedure in pseudo-codice:

```

procedura proc1 (array  $T$ , array  $S$ )
1.    $i \leftarrow 1$ 
2.   while ( $i \leq |T|$ ) do
3.      $j \leftarrow 1$ 
4.     while ( $j \leq |S|$ ) do
5.       Istr
6.        $j \leftarrow j + 1$ 
7.      $i \leftarrow i + 1$ 

procedura mainProc (array  $T$ , array  $S$ , indici  $i$  e  $f$ )
1.   if ( $i < f$ ) then
2.      $numElem \leftarrow f - i + 1$ 
3.     mainProc( $T$ ,  $S$ ,  $i$ ,  $i + numElem/4$ )
4.     mainProc( $T$ ,  $S$ ,  $i + numElem/4$ ,  $i + numElem/2$ )
5.     mainProc( $T$ ,  $S$ ,  $f - numElem/4$ ,  $f$ )
6.     proc1( $T$ ,  $S$ )

```

Gli indici passati nella chiamata iniziale della procedura `mainProc` sono $i = 1$ e $f = n$. Sapendo che `Istr` è un blocco di istruzioni il cui tempo di esecuzione è $O(1)$, studiare il tempo di esecuzione di `mainProc` nel caso peggiore. Per semplicità di analisi, è possibile non considerare gli effetti dovuti agli arrotondamenti (parti intere inferiori e superiori).

Esercizio 3. Definire gli alberi binari di ricerca e gli alberi AVL. Discuterne le proprietà caratterizzanti e le differenze.

Esercizio 4. Sia $G = (V, E)$ un grafo non orientato rappresentato tramite liste di adiacenza. Il grado a distanza due di un nodo è definito come la somma dei gradi dei vicini del nodo stesso.

1. Progettare un algoritmo che calcoli due array, deg e $deg2$, tali che $deg[u]$ contenga il grado del nodo u , e $deg2[u]$ contenga il grado a distanza due del nodo u .
2. Progettare un algoritmo che calcoli l'array $deg2$ dei gradi a distanza due senza mantenere esplicitamente in un array di appoggio il grado dei nodi.

Analizzare il tempo di esecuzione degli algoritmi proposti.

Solo per studenti esonerati: ripetere l'Esercizio 4 assumendo che il grafo sia rappresentato tramite matrice di adiacenza.

NOME	COGNOME
------	---------

Soluzione Esercizio 1:

ALGORITMI I (A.A. 2008-2009)

DOCENTE: IRENE FINOCCHI

APPELLO DEL 22 GENNAIO 2009

NOME	COGNOME
------	---------

Soluzione Esercizio 2:

ALGORITMI I (A.A. 2008-2009)

DOCENTE: IRENE FINOCCHI

APPELLO DEL 22 GENNAIO 2009

NOME	COGNOME
------	---------

Soluzione Esercizio 3:

ALGORITMI I (A.A. 2008-2009)

DOCENTE: IRENE FINOCCHI

APPELLO DEL 22 GENNAIO 2009

NOME	COGNOME
------	---------

Soluzione Esercizio 4:

ALGORITMI I (A.A. 2008-2009)

DOCENTE: IRENE FINOCCHI

APPELLO DEL 22 GENNAIO 2009