

Introduzione agli Algoritmi (secondo canale, A.A. 2010-2011)
Prova Intermedia del 28 Aprile 2011
Prof.ssa Irene Finocchi

TEMPO CONCESSO: 2H.

CONSEGNATE UN FOGLIO PER ESERCIZIO, PENA L'ANNULLAMENTO DEL COMPITO.
COME INTESAZIONE DI CIASCUN FOGLIO, SCRIVETE NOME, COGNOME, E NUMERO DI
ESERCIZIO. NEL CASO IN CUI NON RIUSCIATE A SVOLGERE UN ESERCIZIO, CONSEGNATE
COMUNQUE UN FOGLIO CONTENETE L'INTESAZIONE, MA SENZA SOLUZIONE.
E' AMMESSO L'USO DI LIBRI ED APPUNTI, MA NON DI PORTATILI E TELEFONI CELLULARI.

Esercizio 1 (punti 11/30)

Sia S un array di n valori reali (sia positivi che negativi). Due elementi in S formano un'*inversione a distanza 2* se occupano due posizioni i e $i+2$ nell'array e risulta $S[i] > S[i+2]$. Vogliamo contare il numero di inversioni a distanza 2. Non è difficile progettare un semplice algoritmo iterativo che risolve il problema in tempo $O(n)$. Progettare un algoritmo ricorsivo per questo problema che ottenga anch'esso tempo di esecuzione lineare. Oltre ad una spiegazione a parole dell'algoritmo, darne lo pseudocodice ed analizzarne il tempo di esecuzione.

Esercizio 2 (punti 11/30)

a. Valutare il tempo di esecuzione del seguente frammento di pseudocodice:

```
i=1
while i<n do
  A[i] = A[i]*B[i]
  i=i*2
  j=n
  while j>1 do
    A[j] = B[i]
    j=j-3
```

b. Se $\text{Sub}(j)$ richiede tempo $\Theta(n^3)$, qual è il tempo di esecuzione di questo ciclo?

```
for i=1 to n do
  j=1
  while j<n do
    Sub(j)
    j=j*2
```

Introduzione agli Algoritmi (secondo canale, A.A. 2010-2011)
Prova Intermedia del 28 Aprile 2011
Prof.ssa Irene Finocchi

Esercizio 3 (punti 11/30)

Valutare il tempo di esecuzione dell'algoritmo ricorsivo `Strange` sotto le seguenti ipotesi:

- l'array `A` ha lunghezza `n`;
- nella chiamata iniziale `i=1` e `j=n`;
- la funzione `Fun`, quando invocata su `k` elementi, ha tempo di esecuzione $\Theta(k^3)$.

```
strange(array A, int i, int j)
  if i < j then
    m = (i+j)/2
    q= 4*Strange(A,i,m) + Strange(A,m+1,j) - Strange (A,i,m)
    return Fun(A,i,j,q)
  else return 1
```

Risolvere la relazione di ricorrenza ottenuta per iterazione o usando l'albero della ricorsione.