

**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) e sia  $x$  un valore reale. Vogliamo determinare se ci sono in  $S$  due interi la cui somma è uguale a  $x$ . Non è difficile progettare un semplice algoritmo iterativo che risolve il problema in tempo  $O(n^2)$ , ma questa soluzione non è ottima. Progettare un algoritmo ricorsivo per questo problema che abbia tempo di esecuzione  $O(n \log n)$ . 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=1
  while j<n do
    A[j] = B[i]
    j=j*2
```

- b. Se  $\text{Sub}(j)$  richiede tempo  $\Theta(n^2)$ , qual è il tempo di esecuzione di questo ciclo?

```
for i=1 to n do
  j=n
  while j>1 do
    Sub(j)
    j=j/3
```

**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  $\Theta(k^2)$ .

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

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