

INTRODUZIONE AGLI ALGORITMI, A.A. 2009-2010
(Proff. Calamoneri, Fachini, Finocchi)
Prova Intermedia del 29-4-2010

Esercizio 1 (punti 8)

Sia S un array di n interi. Vogliamo determinare se ci sono almeno due interi uguali in S . 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 e discuterne il tempo di esecuzione.

Esercizio 2 (punti 8)

Sia dato il seguente algoritmo, che elabora una matrice A di n righe e m colonne:

```
elaboraMatrice (A, n, m)
  for j ← 1 to m do
    for i ← 1 to n do A[1][j] ← A[1][j] + A[i][j]
    if A[1][j] ≠ 0 then
      for k ← 1 to j do A[1][j] ← A[1][j] + A[1][k]
```

1. Calcolare il tempo di esecuzione, come funzione sia di n che di m .
2. E' corretto dire che il tempo di esecuzione è $\Theta(m^2)$ nei tre casi: a) $n = O(m)$; b) $n = \Theta(m)$; c) $n = \Omega(m)$? Discutere i tre casi separatamente motivando le risposte.

Esercizio 3. (punti 8)

Qual è il tempo di esecuzione della seguente funzione ricorsiva? Scrivere la relazione di ricorrenza e risolverla:

```
test(n)
  if n ≤ 1 then return
  for i ← 1 to 3 do test(n-1)
```

Esercizio 4 (punti 8)

Sia dato il seguente algoritmo di ordinamento che lavora su un array A di n elementi:

```
algoritmo unAltroSort(A, n)
  BubbleSort(A, n-1)
  Merge(A, 1, n-1, n)
```

dove la subroutine `BubbleSort` è richiamata sui primi $n-1$ elementi, e la subroutine `Merge` fonde il sotto-array $A[1] \dots A[n-1]$ con il sotto-array $A[n]$ contenente un solo elemento.

1. Disegnare l'albero di decisione per $n=3$.
2. Qual è il numero di confronti eseguiti dall'algoritmo nel caso peggiore per $n=3$?
3. Si dia una stima del tempo di esecuzione nel caso peggiore, per valori di n arbitrari.