Informatica Generale
Primo Esonero

docenti: Tiziana Calamoneri, Ivano Salvo
Sapienza Università di Roma

24 aprile 2012

Esercizio 1 (10+2 punti) Si consideri lo pseudocodice del seguente algoritmo di ordinamento. La funzione viene richiamata la prima volta con $i = 1$ e $j = n$.

```
function unAltroSort(A[], i, j)
  if (A[i] > A[j]) then
        scambia(A[i], A[j])
  if (i + 1 ≥ j)) then
        return
  k ← ⌊(j − i + 1) / 3⌋
  unAltroSort(A, i, j − k)   /* si ordinano i primi 2/3 del vettore */
  unAltroSort(A, i + k, j)   /* si ordinano gli ultimi 2/3 del vettore */
  unAltroSort(A, i, j − k)
                  /* si ordinano di nuovo i primi 2/3 del vettore */
```

1. (**3 punti**) si scriva l'equazione di ricorrenza che descrive la complessità computazionale della funzione nel caso peggiore, dandone giustificazione;

2. (**2+5 punti**) si risolva l'equazione trovata usando il teorema principale ed un altro metodo e, per ciascuno, si mostri il procedimento usato;

3. Facoltativo:(**2 punti**) si confronti il tempo di esecuzione del caso peggiore di *unAltroSort* con quello dell'*InsertionSort* e del *MergeSort*, facendo opportune considerazioni sull'efficienza dell'algoritmo proposto.

Esercizio 2 (10 punti) Si progetti un algoritmo ricorsivo che, preso in input un vettore a di interi, lo restituisca modificato in modo tale che ogni valore inizialmente negativo sia sostituito da 0. Ad esempio, se l'input fosse $a = [1, 2, -3, -2, 4]$ l'output sarebbe $a = [1, 2, 0, 0, 4]$.

1. (**2 punti**) si scriva l'algoritmo progettato a parole;
2. (**4 punti**) si scriva lo pseudocodice;
3. (**4 punti**) si valuti la complessità della soluzione tramite un'equazione di ricorrenza.

Esercizio 3 (10+4 punti) Diciamo che un indice k di un vettore di interi a lungo n ($0 \leq k < n$) è il *baricentro* di a, se la somma degli elementi di a prima di k è uguale alla somma degli elementi a cominciare da k fino ad n. Formalmente: $\sum_{i=0}^{k-1} a[i] = \sum_{i=k}^{n-1} a[i]$.

1. (**6 punti**) Si scriva una funzione C `int baricentro(int a[], int n, int *k)` che restituisce 1 se esiste almeno un baricentro nel vettore `a` e 0 altrimenti. Quando torna 1, essa carica nel parametro `k` l'indice del baricentro trovato.

   Si valuti la complessità della soluzione.

2. (**4 punti**) Sotto la precondizione che gli elementi del vettore siano tutti positivi, è possibile migliorare l'efficienza della funzione (anche senza migliorare la complessità asintotica del caso pessimo)? Motivare la risposta (eventualmente con il codice di una funzione `baricentroPos` che sfrutta questa precondizione).

3. Facoltativo: (**4 punti**) Scrivere una funzione ricorsiva che risolve lo stesso problema con un'*unica* scansione del vettore [**Sugg:** scrivere una funzione ausiliaria con parametri aggiuntivi e ricordare che la scansione ricorsiva di un vettore, di fatto, percorre il vettore 2 volte, all'andata e al ritorno dalle chiamate ricorsive].

In tutte le funzioni dell'esercizio 3 verrà apprezzato se il codice sarà corredato di opportune asserzioni logiche (in particolare precondizioni e invarianti dei cicli).