Introduzione agli Algoritmi

Prova intermedia del 16 aprile 2013

Proff. Emanuela Fachini ed Irene Finocchi

Ogni algoritmo deve essere specificato con precisione, definendo l'input, l'output, eventuali precondizioni e postcondizioni. L'eventuale pseudocodice deve inoltre essere ben commentato. Le risposte non motivate o non sufficientemente chiare non saranno prese in considerazione.

Tempo concesso: 2h 30m

Problema 1

Definire la relazione di ricorrenza che esprime il tempo di esecuzione T(n) del seguente algoritmo:

```
algoritmo analizzami (n: intero)
  if n<3 return;
  i = 1
  while (i ≤ n) do i = i*2;
  analizzami(n-1)
```

Dimostrare (canale AL: per sostituzione; canale MZ: per iterazione) che la relazione di ricorrenza ha soluzione T(n) = O(n logn).

Problema 2

Progettare un algoritmo che, dato un array S di n interi, restituisca `true` se è possibile partizionare S in coppie di elementi che hanno tutte lo stesso valore totale (il valore totale di una coppia è la somma dei due elementi della coppia).

Si possono utilizzare gli algoritmi visti durante il corso.

Esempi

- L'array S = [4, 5, 1, 3, 7, 4] è partizionabile nel modo specificato sopra: le coppie in cui possiamo partizionare S sono (5,3),(1,7),(4,4).
- L'array S = [1, 1, 2, 1] non è invece partizionabile, e l'algoritmo deve restituire `false`.

Analizzare il tempo di esecuzione e la correttezza dell'algoritmo proposto (canale AL: per la progettazione di cicli si definisca un'invariante e la si utilizzi per dimostrarne la correttezza).

Problema 3

Si supponga di scrivere una variante di MergeSort, chiamata MergeSort-K che, invece di suddividere l'array da ordinare in 2 parti (e ordinarle separatamente), lo suddivide in K parti, le ordina ognuna riapplicando MergeSort-K, e le riunifica usando un'opportuna variante di Merge (che fonde K sottoarray invece di 2).

1. Quanto tempo si richiede per trovare il minimo tra K elementi qualunque?

2. Quanto tempo si richiede per fondere K sottoarray ordinati?

3. Se K = 4, come cambia – *se cambia* – il tempo di esecuzione di MergeSort-K rispetto a quello di MergeSort?

Non occorre dare lo pseudocodice di MergeSort-K, ma solo giustificare adeguatamente tutte le risposte.

Facoltativo. Se avete risolto i punti 1 - 3, rispondete alle seguenti domande *assumendo che* $K = \sqrt{n}$:

4. Quale sarebbe l'equazione di ricorrenza che descrive il tempo di esecuzione di MergeSort-K con K = √n? E' sufficiente impostare la ricorrenza senza risolverla.

5. Il tempo di esecuzione sarebbe ancora ottimo? Perché? (Non occorre risolvere la ricorrenza per rispondere a questa domanda.)