

INTRODUZIONE AGLI ALGORITMI

Esame Scritto a canali unificati

docenti: T. CALAMONERI, A. MONTI
Sapienza Università di Roma
22 Giugno 2021

Esercizio 1 (10 punti): Si consideri la seguente funzione:

```
funzione exam( $n, k$ : interi)
  prod  $\leftarrow$  1
   $i \leftarrow k$ 
  while  $i \geq 1$  do
    prod  $\leftarrow$  prod *  $i$ 
    if  $i$  pari then  $i \leftarrow i/2$ 
    else  $i \leftarrow (i - 1)/2$ 
  if  $n = 1$  return prod
  return prod * exam( $n - 1, k$ )
```

Si imposti la relazione di ricorrenza che ne definisce il tempo di esecuzione facendo particolare attenzione al caso base; si giustifichi l'equazione ottenuta, e la si risolva usando il metodo iterativo ed il metodo di sostituzione, commentando opportunamente i passaggi del calcolo.

Esercizio 2 (10 punti): Data una sequenza S di n bit memorizzata in un array A , progettare un algoritmo che ordina S in tempo $\Theta(n)$. L'algoritmo deve ordinare *in loco*.

Dell'algoritmo proposto si dia la descrizione a parole, si scriva lo pseudocodice e si calcoli il costo computazionale.

Per quale motivo è possibile ordinare S in tempo lineare?

Esercizio 3 (10 punti): Dato un albero binario T , radicato nel nodo r , definiamo *altezza minimale* di T la minima distanza (cioè il minimo numero di archi) tra r e una qualsiasi foglia di T .

Progettare un algoritmo che, dato il puntatore alla radice di un albero binario memorizzato tramite record e puntatori, restituisca la sua altezza minimale.

Il costo dell'algoritmo deve essere $O(n)$, dove n è il numero di nodi dell'albero.

Dell'algoritmo proposto si dia la descrizione a parole, si scriva lo pseudocodice e si motivi il costo computazionale.

Quali sono i valori minimo e massimo che l'altezza minimale di T può assumere? Motivare la risposta.

Idee per la soluzione

L'equazione di ricorrenza dell'esercizio 1 è:

$$T(n, k) = T(n - 1, k) + \Theta(\log k) \text{ e } T(1, k) = \Theta(\log k).$$

Errata ogni funzione in cui compaia solo n , solo k , o altri parametri, come ad esempio i .

Risolvendo, si ottiene la soluzione $T(n, k) = \Theta(n \log k)$.

Per risolvere l'esercizio 2 si può usare il Counting sort senza dati satellite, visto che l'array A contiene solo due valori, 0 e 1. Alternativamente, è possibile implementare una leggera modifica della funzione `Partiziona` del Quicksort, per spostare a sinistra tutti gli zeri ed a destra tutti gli uni.

È possibile ottenere un algoritmo lineare perché l'array contiene un numero costante, cioè $O(n)$, di valori.

Per risolvere l'esercizio 3 basta implementare una modifica della visita in post-order simile a quella necessaria per calcolare l'altezza di un albero, ma nella quale viene passato alla funzione chiamante il minimo dei valori ritornati dai figli anziché il massimo.