Esercitazione

29 marzo 2022

Tipologie di esercizi

1. Formalizzazione problema computazionale
2. Notazioni asintotiche:
   a) Via definizione (c, n_0)
   b) Applicando le proprietà
   c) Sequenze di funzioni da ordinare
   d) Confronto tempo di esecuzione di algoritmi
3. Calcolo del tempo di esecuzione di algoritmi (senza chiamate ricorsive)
4. Scrivere la relazione di ricorrenza per il tempo di esecuzione di algoritmi ricorsivi
5. Risolvere relazioni di ricorrenza
6. Tecnica del Divide et Impera

Altre relazioni di ricorrenza

Abbiamo considerato una sotto-famiglia

- $T(n) = qT(n/2) + cn$ con $T(2)=c$

  per $q=1$ allora $T(n)= \Theta(n)$

  $q=2$ allora $T(n)= \Theta(n \log_2 n)$

  $q>2$ allora $T(n)= O(n^{\log_2 q})$

- $T(n)=2T(n/2) +cn^2$

  $T(n)= \Theta(n^2)$

Tempo di esecuzione 4 (svolto)

Il tempo di esecuzione del seguente frammento di pseudocodice è

```
for i=1 to n/2
   PARTITION(A,i,n)
```

A. Θ (log n)
B. Θ (n log n)
C. Θ (n)

D. Nessuna delle risposte precedenti

Merge (svolto)

Quanti confronti effettua l'algoritmo `MERGE` per la fusione dei due array ordinati [1,5,6,7] e [2,3,4]?

A. 4 B . 6 C. 12 D. Nessuna delle risposte precedenti

```
i = 1, j = 1
while (both lists are nonempty) {
   if (a_i ≤ b_j) append a_i to output list and increment i
   else(a_i > b_j)append b_j to output list and increment j
}
append remainder of nonempty list to output list
```

Relazione di ricorrenza 2 (soluzione)

- Si consideri la seguente relazione di ricorrenza.

$$T(0) = 1$$
$$T(1) = 3$$
$$T(n) = T(n - 2) + n$$

Quanto valgono $T(6)$ e $T(9)$?

- Risolvere la relazione di ricorrenza con tutti i metodi possibili.

Discussione di una soluzione proposta da uno studente sulla piattaforma.
Soluzione con unrolling e alberi di ricorsione.

Dalla piattaforma

- (Relazioni di ricorrenza 2)

Nella risoluzione della relazione di ricorrenza

$$\mathbf{T(n) = 2\,T(\,n/2) + n}, \text{ con } T(1) = c$$

col metodo di iterazione, qual è il valore di T(n) alla i-esima iterazione? (già svolto)

- (Soluzione relazione di ricorrenza 3)

La soluzione della relazione di ricorrenza

$T(n) = 2T(n/2) + c$, con $T(1) = c$, è: … (da svolgere)

- (Soluzione relazione di ricorrenza 4)

La soluzione della relazione di ricorrenza

$T(n) = 4T(n/2) + n$, con $T(1) = 1$, è: (segue)

Soluzione relazione di ricorrenza 4 (risolto)

La soluzione della relazione di ricorrenza

T(n) = 4T(n/2) + n, con **T(1) = 1**, è:

Soluzione relazione di ricorrenza 4

Scelte

Opzioni scelta	T(n) = Theta(n^(log4 2)) ☐	T(n) = Theta(n) ☐	T(n) = Theta(n^2) ☐	Nessuna delle risposte precedenti ☐
Numero di risposte	1	5	5	5

Massimo con D&I

Descrivere ed analizzare un algoritmo **basato sulla tecnica Divide et Impera** che dato un array A[1, …, n] di interi ne restituisca il massimo.

Discussione di due soluzioni proposte da studenti sulla piattaforma

Altri esempi

Sia $T(1) = 1$. Valutate

- $\boxed{T(n) = 2T(n/2) + n^3}$
- $T(n) = T(9n/10) + n$
- $T(n) = 16T(n/4) + n^2$
- $T(n) = 7T(n/3) + n^2$
- $T(n) = 7T(n/2) + n^2$
- $T(n) = 2T(n/3) + \sqrt{n}$
- $\boxed{T(n) = T(n-1) + n}$
- $\boxed{T(n) = T(\sqrt{n}) + 1}$

Alcuni esercizi sulla tecnica del Divide et Impera.

1. a) Descrivere gli aspetti essenziali della tecnica Divide et Impera, utilizzando lo spazio designato.

   b) Descrivere ed analizzare un algoritmo **basato sulla tecnica Divide et Impera** che dato un array $A[1, ..., n]$ di interi ne restituisca il massimo.

2. Sia $V[1..n]$ un vettore ordinato di 0 e 1.

   Descrivere ed analizzare un algoritmo per determinare il numero di 0 presenti in $V[1..n]$ in tempo $O(\log n)$.

5. a) Fornire lo pseudocodice di un algoritmo ricorsivo che ordini un array $A[1..n]$ nel seguente modo: prima ordina ricorsivamente $A[1..n-1]$ e poi inserisce $A[n]$ nell'array ordinato $A[1..n-1]$.

   b) Analizzare la complessita' di tempo dell'algoritmo proposto al punto b).

6. Sia dato un vettore binario ordinato $A[1..n]$.

   (a) Progettare un algoritmo di complessita' $\Theta(n)$ nel caso peggiore, che conti il numero di occorrenze di 1 nel vettore A.

   (b) Progettare un algoritmo di complessita' $O(\log n)$, che conti il numero di occorrenze di 1 nel vettore A.

7. Sia dato un vettore ordinato $A[1..n]$ di interi distinti. Progettare un algoritmo che determini, in tempo $O(\log n)$, se esiste o meno un intero i tale che $A[i] = i$.

8. Descrivere ed analizzare un algoritmo basato sul paradigma *divide et impera* che dato un vettore *ordinato* $A[1..n]$ di interi strettamente positivi (cioe' per ogni $1 \leq i \leq n$, $A[i] \geq 1$), restituisca il numero di occorrenze di 1 nel vettore A. L'algoritmo deve avere complessita' di tempo $O(\log n)$.

Occorrenze consecutive di 2 (D&I) (dalla piattaforma)

Si scriva lo pseudo-codice di un algoritmo ricorsivo basato sulla tecnica Divide et Impera che prende in input un array di interi positivi e restituisce il massimo numero di occorrenze **consecutive** del numero ‘2’.

Ad esempio, se l'array contiene la sequenza <2 2 3 6 2 2 2 2 3 3> allora l'algoritmo restituisce 4. Occorre specificare l'input e l'output dell’algoritmo.

Discussione con la classe. Siete invitati a inserire la soluzione corretta sulla piattaforma