Algoritmi e Strutture Dati

Foglio 2
06/03/2023

Esercizio 1. Modificare la subroutine MERGE di MERGE-SORT in modo da non utilizzare le sentinelle.

Esercizio 2. Scrivere uno pseudocodice per l'algoritmo di ricerca binaria visto a lezione e dimostrarne la correttezza.

Esercizio 3. Scrivere uno pseudocodice per l'algoritmo di elevamento a potenza visto a lezione e dimostrarne la correttezza.

Esercizio 4. Sia $T(n)$ il numero di righe stampate dal seguente algoritmo ricorsivo. Dare un limite asintotico stretto per $T(n)$.

```
function f(n)

if n > 1:
    print(``still going'')
    f(n/2)
    f(n/2)
```

Esercizio 5. Usando il metodo di sostituzione, dimostrare le seguenti (si assuma in tutti i casi che $T(1) = c$):

1. La soluzione di $T(n) = T(n-1) + n$ è $O(n^2)$;
2. La soluzione di $T(n) = 2T(\lfloor n/2 \rfloor + 17) + n$ è $O(n \log n)$.

Esercizio 6. Utilizzare un albero di ricorsione per determinare un buon limite asintotico superiore per $T(n)$ e applicare il metodo di sostituzione per verificarlo (si assuma in tutti i casi che $T(1) = c$):

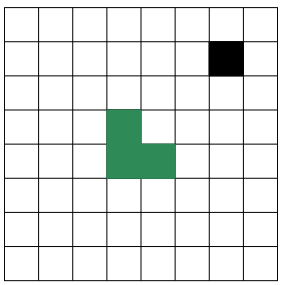
1. $T(n) = 4T(n/2 + 2) + n$;
2. $T(n) = 2T(n-1) + 1$;
3. $T(n) = 2T(n/2) + n \log n$.

Esercizio 7. Consideriamo il problema di calcolare la somma degli elementi di un array di n numeri. Usando il Divide et Impera, fornire un algoritmo che risolva il problema e stimarne la complessità. E' migliore dell'algoritmo brute force che scorre l'array?

Esercizio 8. Ci è dato un vettore di n numeri e notiamo che alcuni di essi si ripetono. Come possiamo rimuovere tutti i doppioni in $O(n \log n)$ passi?

Esercizio 9. Si vuole tassellare una scacchiera $n \times n$ (con n potenza di 2), priva di una casella (nera in figura), con dei tasselli a forma di L occupanti tre caselle (tassello verde in figura). Fornire un algoritmo che risolva il problema in $O(n^2)$ passi.

Suggerimento: Si disponga il primo tassello come in figura.



Esercizio 10. Utilizzate le seguenti idee per sviluppare un algoritmo non ricorsivo in tempo lineare per risolvere il problema del massimo sottoarray. Iniziate dall'estremità sinistra dell'array e procedete verso destra, registrando il massimo sottoarray trovato fino a quel punto. Conoscendo un massimo sottoarray di $A[1..j]$, aggiornate la soluzione considerando il massimo sottoarray che termina con l'indice $j + 1$ sulla base della seguente osservazione: un massimo sottoarray di $A[1..j + 1]$ è un massimo sottoarray di $A[1..j]$ o un sottoarray di $A[i..j + 1]$, per $1 \leq i \leq j + 1$. Determinate un massimo sottoarray della forma $A[i..j + 1]$ in tempo costante, supponendo di conoscere un massimo sottoarray che termina con l'indice j.