

Algoritmi e Strutture Dati

13 Luglio 2016

Cognome Nome Matricola

Note

1. La leggibilità è un prerequisito: parti difficili da leggere potranno essere ignorate.
2. Quando si presenta un algoritmo è fondamentale spiegare l'idea sottostante e motivarne la correttezza.
3. L'efficienza è un criterio di valutazione delle soluzioni proposte.
4. Si consegnano tutti i fogli, con nome, cognome, matricola e l'indicazione *bella copia* o *brutta copia*.

Domande

Domanda A (4 punti) Sia data la seguente equazione di ricorrenza:

$$T(n) = 5T(\lfloor n/3 \rfloor) + 2n^2$$

Si fornisca un limite asintotico stretto per la soluzione.

Domanda B (4 punti) Si consideri una tabella hash di dimensione $m = 9$, e indirizzamento aperto con doppio hash basato sulle funzioni $h_1(k) = k \bmod m$ e $h_2(k) = 1 + k \bmod (m - 2)$. Si descriva in dettaglio come avviene l'inserimento della sequenza di chiavi: 12, 3, 22, 14, 38.

Domanda C (5 punti) Dato un albero nel quale i nodi contengono una chiave, si definisca *costo* di un cammino dalla radice ad una foglia, come la somma delle chiavi dei nodi che compaiono nel cammino. Scrivere una funzione `MaxPath(T)` che opera nel modo seguente. Prende in input un albero binario T , con radice $T.root$, e nodi x che hanno come campi $x.k$, $x.l$ e $x.r$, ovvero una chiave, il puntatore al figlio sinistro e destro, rispettivamente. Resituisce la il costo del cammino di costo massimo dalla radice ad una foglia. Valutarne la complessità.

Esercizi

Esercizio 1 (7 punti) Si consideri una struttura che chiamiamo `OderedStack` con le seguenti operazioni:

- `OEmpty(S)`: Ritorna un booleano che dice se la struttura è vuota
- `OPop(S)`: Estrae l'ultimo elemento inserito
- `OTop(S)`: Ritorna il valore dell'ultimo elemento inserito
- `OPush(S,x)`: Inserisce nello stack x , eliminando prima ogni elemento che sia maggiore di x .

Utilizzando una struttura dati stack, fornire una implementazione della suddetta struttura ovvero

- i. Specificare come è definita la struttura dati e fornire lo pseudocodice dei metodi elencati.
- ii. Mostrare che una sequenza di n operazioni, a partire da struttura vuota, ha costo ammortizzato $O(n)$ (e quindi ogni singola operazione ha costo ammortizzato $O(1)$).

Esercizio 2 (11 punti) Data una sequenza di numeri $X = x_1 x_2 \dots x_n$ si vuole determinare una sottosequenza $x_{i_1} x_{i_2} \dots x_{i_k}$ di X , crescente, ovvero tale che $x_{i_1} \leq x_{i_2} \leq \dots \leq x_{i_k}$, di lunghezza massima. Ad esempio, se $X = 77 \ 69 \ 70 \ 19 \ 71 \ 25 \ 52 \ 26 \ 28 \ 64$, una sottosequenza crescente di lunghezza massima è 19 25 26 28 64

- i. fornire una caratterizzazione ricorsiva della lunghezza l_i di una sottosequenza crescente di lunghezza massima che abbia come primo carattere x_i ;
- ii. tradurre tale definizione in un algoritmo `LIS(X,n)` (bottom up o top down con memoization) che determina la lunghezza di una sottosequenza crescente di lunghezza massima della sequenza $X[1..n]$;
- iii. trasformare l'algoritmo in modo individui anche la sottosequenza, non solo la sua lunghezza;
- iv. valutare la complessità dell'algoritmo.