

Algoritmi e Strutture Dati

27 Agosto 2020

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 e motivarne la correttezza.
3. L'efficienza e l'aderenza alla traccia sono criteri di valutazione delle soluzioni proposte.
4. Si consegna la scansione dei fogli di bella copia, e come ultima pagina un documento di identità.

Domande

Domanda A (8 punti) Si consideri la seguente funzione ricorsiva che dato un array A e due indici $1 \leq p \leq r \leq A.length$ restituisce la somma degli elementi in $A[p, r]$

```
sum(A, p, r)
  if p < r
    q = (p+r)/2
    return sum(A, p, q) + sum(A, q+1, r)
  else
    return A[p]
```

Dimostrare induttivamente che la funzione è corretta, ovvero che $sum(A, p, r)$ ritorna $\sum_{i=p}^r A[i]$. Inoltre, determinare la ricorrenza che esprime la complessità della funzione e risolverla con il Master Theorem.

Domanda B (5 punti) (i) Si spieghi brevemente come funzionano le tabelle hash con open addressing. (ii) Data una tabella di dimensione m , la funzione di hash $h(k, i) = (k + 2 * i) \bmod m$ è appropriata? Si motivi la risposta.

Esercizi

Esercizio 1 (9 punti) Realizzare un arricchimento degli alberi binari di ricerca che permetta di ottenere per ogni nodo x , in tempo costante, la media dei valori memorizzati nel sottoalbero radicato in x .

Indicare quali campi occorre aggiungere ai nodi. Fornire il codice per la funzione `avg(x)` che restituisce la media delle chiavi nel sottoalbero radicato in x e la procedura di inserimento di un nodo `insert(T, z)`.

Esercizio 2 (9 punti) Lungo una strada ci sono, in vari punti, n parcheggi liberi e n auto. Un posteggiatore ha il compito di parcheggiare tutte le auto, e lo vuole fare minimizzando lo spostamento totale da fare. Formalmente, dati n valori reali p_1, p_2, \dots, p_n e altri n valori reali a_1, a_2, \dots, a_n , che rappresentano le posizioni lungo la strada rispettivamente di parcheggi e auto, si richiede di assegnare ad ogni auto a_i un parcheggio $p_{h(i)}$ minimizzando la quantità

$$\sum_{i=1}^n |a_i - p_{h(i)}|.$$

1. Si consideri il seguente algoritmo greedy. Si individui la coppia (auto, parcheggio) con la minima differenza. Si assegni quell'auto a quel parcheggio. Si ripeta con le auto e i parcheggi restanti fino a quando tutte le auto sono parcheggiate. Dimostrare che questo algoritmo non è corretto, esibendo un controesempio.
2. Si consideri il seguente algoritmo greedy. Si assuma che i valori p_1, p_2, \dots, p_n e a_1, a_2, \dots, a_n siano ordinati in modo non decrescente. Si produca l'assegnazione $(a_1, p_1), (a_2, p_2), \dots, (a_n, p_n)$. Dimostrare la correttezza di questo algoritmo per il caso $n = 2$.