Algoritmi e Strutture Dati
14 Settembre 2017

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 soggiacente e motivarne la correttezza.
3. L'efficienza è un criterio di valutazione delle soluzioni proposte.

Consegna (1 punto) Consegnare tutti i fogli, con nome, cognome, matricola e l'indicazione esplicita *bella copia* o *brutta copia*.

Domande

Domanda A (4 punti) Dare una soluzione asintotica per la ricorrenza $T(n) = 3T(n/2) + n(n+1)$.

Domanda B (4 punti) Si consideri una tabella hash di dimensione $m = 8$, gestita mediante chaining (liste di trabocco) con funzione di hash $h(k) = k \ mod \ m$. Si descriva in dettaglio come avviene l'inserimento della sequenza di chiavi: 14, 10, 22, 18, 19.

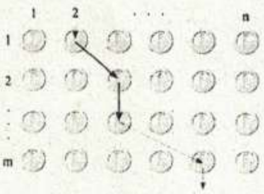
Domanda C (5 punti) Si consideri una variante degli alberi binari di ricerca nella quale i nodi `x` hanno un campo `x.pred` (predecessore) invece che il campo `x.p` (parent). Realizzare la procedura di `Insert(T,z)` che inserisce un nodo `z` nell'albero. Valutarne la complessità.

Esercizi

Esercizio 1 (7 punti) Sia `A[1..n]` un array di interi distinti ordinato in senso crescente. Dimostrare che dato un qualunque indice `i`, se `A[i] > i` allora `A[j]>j` per ogni `j>i` e analogamente se `A[i] < i` allora `A[j]<j` per ogni `j<i`.

Utilizzare l'osservazione per realizzare una funzione `Fix(A)` che dato l'array di interi `A[1..n]` ordinato senza ripetizioni restituisce un indice `i` tale che `A[i]=i`, se esiste, e `0` altrimenti. Valutarne la complessità.

Esercizio 2 (9 punti)



Mario deve attraversare una griglia come in figura, dall'alto verso il basso e per farlo, ad ogni passo salta verso il basso di una riga, spostarsi contestualmente a destra di quanto vuole. Un esempio di attraversamento è indicato in figura. Ogni casella contiene una moneta di un certo valore (possibilmente negativo) per cui nell'attraversamento Mario totalizzerà un certo guadagno.

Supponendo che la griglia abbia dimensione $m \times n$ e che per $i \in \{1, \ldots, m\}$, $j \in \{1, \ldots, n\}$ la casella (i, j) contenga una moneta di valore $V[i, j]$ realizzare un algoritmo che identifica un attraversamento di valore massimo.

Più precisamente:

i. dare una caratterizzazione ricorsiva del guadagno massimo $G[i, j]$ di un attraversamento della sottogriglia $[i..m, j..n]$;

ii. usare la caratterizzazione al punto precedente per ottenere un algoritmo `mario(V,m,n)` che dato l'array `V` con il valore delle monete determini il guadagno massimo di un attraversamento;

iii. trasformare l'algoritmo in modo che restituisca oltre al guadagno anche anche l'indicazione dell'attraversamento da seguire;

iv. valutare la complessità dell'algoritmo.

Nota: Correzione, risultati e visione dei compiti: *Lunedì 25 Settembre, ore 14:00 (da confermare)*

A

$T(n) = 3T\left(\frac{n}{2}\right) + n(n+1)$

$a = 3$
$b = 2$
$f(n) = n(n+1)$

$n^{\log_2 3}$ vs $f(n) = n(n+1)$

$1 < \log_2 3 < 2$ $\qquad 0 \leq \varepsilon < \underbrace{2 - \log_2 3}_{>0}$ $\qquad f(n) = \Omega\left(n^{\log_2 3 + \varepsilon}\right)$

3 CASO? verifico $a f\left(\frac{n}{b}\right) \leq k f(n)$ $\forall n \geq n_0$ $\quad 0 < k < 1$

$3 \frac{n}{2} \underbrace{\left(\frac{n}{2} + 1\right)}_{\leq \text{ minore di } \frac{5n}{8} \quad n \geq 8} \leq k\, n(n+1)$

Basta ipotizzare qualsiasi cosa per ottenere un risultato giusto dato che non cerchiamo soluzione esatta.

$\leq \frac{3}{2} n \frac{5n}{8} = \frac{15}{16} n^2 \leq \underbrace{\frac{15}{16}}_{k} n(n+1)$

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

B

$h(k) = k \bmod 8$

$h(14) = 14 \bmod 8$

$h(10) = 10 \bmod 8$

14
10
22
18
19

0		
1		
2	10	→ 18
3	19	
4		
5		
6	14	→ 22
7		

(C)

```
Insert (T, z)
  x = T.root
  y = nil;  w = nil
  while (x ≠ nil)
     y = x
     if z.key ≤ x.key
        x = x.left   w = y
     else
        x = x.right
  if y = nil
     T.root = z
  else if z.key < y.key
        y.left = z ;  [z.pred = y.pred]
     else
        y.right = z
        [z.pred = y.]
```

per aggiornare predecessore di z.

// devo aggiornare il successore di z. che avrà come predecessore z stesso

```
if ~~w~~ ≠ nil
   w.pred = z
```

Es 1

$A[1 \dots n]$ ordinato $\nearrow$ senza ripetizioni (di interi)

$\forall i \quad A[i] > i \Rightarrow \forall j > i \quad A[j] > j$

$A[i] < i \Rightarrow \forall j < i \quad A[j] < j$

$A[i] > i$ allora $A[i] \geq i+1$ (dato che sono interi)

osservo se $i < n$ allora $A[i+1] > A[i]$ (per def di crescente) $\geq i+1$

$\Rightarrow A[i+1] > i+1$

quindi $\underset{\text{induttivo}}{\forall h \geq 1} \quad A[i+h] > i+h$

Fix(A) trova i t.c. $A[i] = i$ se esiste
o
0 altrimenti

```
Fix-rec(A, p, r)
  if p ≤ r
     q = (p+r)/2
     if A[q] > q
        return FixRec(A, p, q-1)
     else if A[q] < q
        return FixRec(A, q+1, r)
     else
        return q
  else
     return 0
```

```
Fix(A)
  return FixRec(A, 1, n)
```

$T(n) = T(\frac{n}{2}) + c$

per Master Theorem $= \Theta(\log n)$

Es 2

$v[i,j]$ = valore delle monete

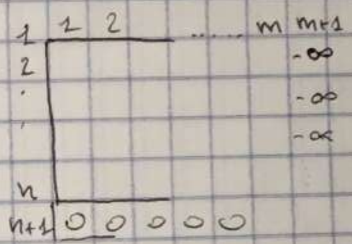
$G[i,j]$ = guadagno max attraversando $[i \dots m, j \dots n]$

$$G_{i,j} = \begin{cases} 0 & \text{se } i = m+1 \quad \text{// non ho niente da attraversare} \\ -\infty & \text{se } j = n+1 \quad \text{// perdo tutto quindi non andrò mai} \\ \max(v[i,j] + G[i+1,j],\ G[i,j+1]) & \end{cases}$$

Mario (V, m, n)

```
allocare G[1..m+1, 1..n+1]  ==>
for i=1 to n
    G[i,n+1] = -∞
for j=1 to n
    G[m+1, j] = 0
for i = m to 1
    for j = n to 1
        if v[i,j] + G[i+1,j] < G[i,j+1]
        then
            G[i,j] = G[i,j+1]; S[i,j] = S[i,j+1]
        else
            G[i,j] = v[i,j] + G[i+1,j]; S[i,j] = j
· return G[1,1]
```

$\Theta(n \cdot m)$



$S[i,j]$ = colonna della casella che realizza l'ottimo

```
// Se interessa il percorso  <-> lascio stare
// S[1,1] = j1 // l'inizio
j = 1
for i = 1 to m
    j = S[i,j]
    print cella "i,j"
```

$\Theta(m)$

Costo complessivo $\Theta(n \cdot m)$