Algoritmi e Strutture Dati
18 Febbraio 2020

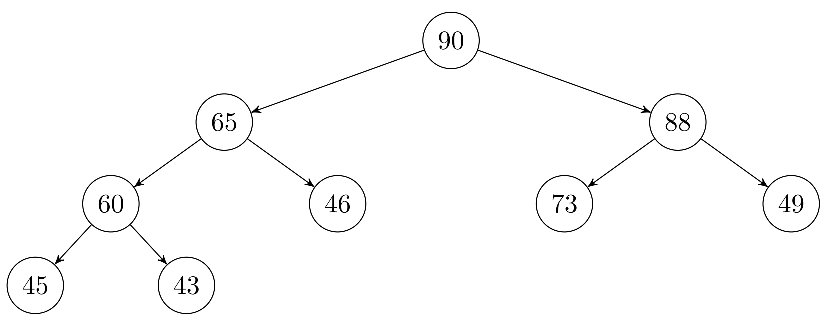
Domande

Domanda A (5 punti) Risolvere la ricorrenza $T(n) = 4T(n/2) + n^2\sqrt{n}$ utilizzando il master theorem.

Soluzione: Rispetto allo schema generale si ha $a = 4$, $b = 2$, $f(n) = n^2\sqrt{n} = n^{\frac{5}{2}}$. Si osserva che $\log_b a = 2$ quindi $f(n) = \Omega(n^{\log_b a+\epsilon})$ (per $0 < \epsilon \leq \frac{1}{2}$). In aggiunta vale la condizione di regolarità, ovvero $af(\frac{n}{b}) \leq cf(n)$ per qualche $c < 1$. Infatti $af(\frac{n}{b}) = 4f(\frac{n}{2}) = 4\frac{n^2}{4}\sqrt{\frac{n}{2}} = n^2\sqrt{\frac{n}{2}} \leq cn^2\sqrt{n}$ per $c \geq \frac{1}{\sqrt{2}}$ (che è < 1). Quindi $T(n) = \Theta(n^2\sqrt{n})$.

Domanda B (4 punti) Dato l'array $A = [60, 90, 49, 65, 46, 73, 88, 45, 43]$, mostrare in forma di albero, il max-heap prodotto dalla procedura `BuildMaxHeap`.

Soluzione: Per la descrizione del processo di costruzione, si rimanda al libro. Si noti che la procedura **non** consiste in una serie di inserimenti. Il risultato è il seguente:



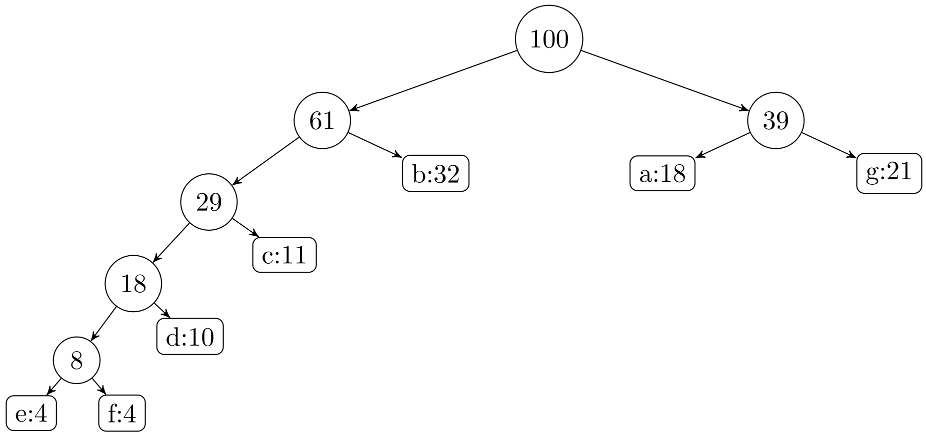
ovvero, in forma di array, $[90, 65, 88, 60, 46, 73, 49, 45, 43]$.

Domanda C (5 punti) Indicare il codice prefisso ottenuto utilizzando l'algoritmo di Huffmann per l'alfabeto $\{a, b, c, d, e, f, g\}$, supponendo che ogni simbolo appaia con le seguenti frequenze.

a	b	c	d	e	f	g
18	32	11	10	4	4	21

Spiegare il processo di costruzione del codice.

Soluzione: Per il processo di costruzione del codice si rimanda al libro. Il risultato è il seguente:



Esercizi

Esercizio 1 (7 punti) Realizzare una procedura `BST(A)` che dato un array `A[1..n]` di interi, ordinato in modo crescente, costruisce un albero binario di ricerca di altezza minima che contiene gli elementi di `A` e ne restituisce la radice. Fornire un'argomentazione che supporti il fatto che l'albero ha altezza minima. Per allocare un nuovo nodo dell'albero si utilizzi una funzione `mknode(k)` che dato un intero `k` ritorna un nuovo nodo con `x.key=k` e figlio destro e sinistro `x.left = x.right = nil`. Valutarne la complessità.

Soluzione:

i. L'implementazione è la seguente:

```
BST(A)
   return BST-rec(A,1,n)

BST-rec(T,A,p,q)
   if p <= q
      m = floor(p+q/2)
      x=mknode(A[m])
      x.l = BST-rec(A,p,m-1)
      x.r = BST-rec(A,m+1,q)
    else
      x = nil

    return x
```

Si può dimostrare, per induzione su n, che l'altezza dell'albero generato è $\lceil \log_2(n+1) \rceil$, quindi è la minima possibile.

ii. Si ottiene la ricorrenza $T(n) = 2T(n/2)+\Theta(1)$. Applicando il master theorem si ottiene quindi come costo $T(n) = O(n)$.

Esercizio 2 (10 punti) Si ricordi che data una sequenza $X = x_1 \dots x_k$, si indica con X_i il prefisso $x_1 \dots x_i$. Una sottosequenza di X è $x_{i_1} \dots x_{i_h}$ con $1 \leq i_1 < i_2 < \dots i_h \leq k$, ovvero è una sequenza ottenuta da X eliminando alcuni elementi. Quando Y è sottosequenza di X si scrive $Y \sqsubseteq X$.

Realizzare un algoritmo che, date due sequenze $X = x_1 \dots x_k$ e $Y = y_1 \dots y_h$ determina una *shortest common supersequence* (SCS) ovvero una sequenza Z, di lunghezza minima, tale che $X \sqsubseteq Z$ e $Y \sqsubseteq Z$. Ad esempio per $X = abf$ e $Y = afgj$ una SCS è $abfgj$.

i. Dare una caratterizzazione ricorsiva della lunghezza $l_{i,j}$ di una SCS di X_i e Y_j e dedurne un algoritmo;

ii. valutare la complessità dell'algoritmo.

Soluzione: La caratterizzazione ricorsiva è:

$$l_{i,j} = \begin{cases} i + j & \text{if } i = 0 \text{ or } j = 0 \\ l_{i-1,j-1} + 1 & \text{if } i, j > 0 \text{ e } x_i = y_j \\ \min\{l_{i,j-1}, l_{i-1,j}\} + 1 & \text{if } i, j > 0 \text{ e } x_i \neq y_j \end{cases}$$

Ne segue l'algoritmo che riceve in input le stringhe, nella forma di array di caratteri $X[1, k]$, $Y[1, h]$ e usa una matrice $L[0..k, 0..h]$ dove $L[i, j]$ rappresenta la lunghezza della minima SCS di X_i e Y_j.

```
SCSlen (X,Y,k,h)
  for i=0 to k
    L[i,0] = i
  for j=1 to h
    L[0,j] = j

  for i=1 to k
    for j = 1 to h
      if (X[i] = X[j])
         L[i,j] = L[i-1,j-1] + 1
      else
         L[i,j] = min (L[i,j-1], L[i-1,j]) + 1

  return L[k,h]
```

La complessità è visto che ci sono due cicli annidati, ripetuti k e h volte, con corpo avente costo costante, $\Theta(hk)$.