CORREZIONE ESAME 3/07/15

(ESERCIZI SVOLTI DAL PROF BALDAN) 1

(A) $\boxed{T(n) = 4T\left(\frac{n}{2}\right) + n^2\sqrt{n}}$

$a=4$, $b=2$, $f(n) = n^2\sqrt{n} = n^{2+\frac{1}{2}}$, $n^{\log_b a} = n^2$

Siamo nel caso 3? $\to$ $f(n) \overset{?}{=} \Omega\left(n^{\log_b a+\varepsilon}\right)$ esiste $\varepsilon>0$

$$\lim_{n\to\infty} \frac{f(n)}{n^{\log_b a+\varepsilon}} > 0 \to \frac{n^{2+\frac{1}{2}}}{n^{2+\varepsilon}} \to 0<\varepsilon<\frac{1}{2}$$

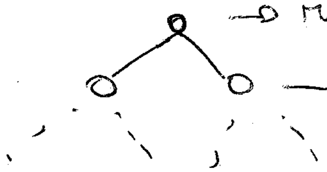
Verifichiamo la condizione di regolarità

$$a f\left(\frac{n}{b}\right) \le c f(n) \qquad c<1$$

$$4\cdot \frac{n^2}{4}\cdot\frac{\sqrt{n}}{2} \le c\, n^2\sqrt{n} \to \boxed{c \ge \frac{1}{\sqrt{2}}}$$

$\boxed{T(n)=\Theta(f(n))}$ $\leftarrow$ SIAMO NEL CASO 3 $\leftarrow$ OK SODDISFATTA LA REGOLARITÀ

(B) Altezza massima B-albero di grado t con n chiavi



→ radice può avere 1 chiave

→ gli altri figli hanno $t-1$ chiavi

Quindi $\boxed{h = \log_t \frac{n+1}{2}}$

(c) Complete (x) // ritorna h dell'albero se completo oppure -1

```
if x = nil
   return 0
else
   h1 = complete(x.left)
   h1 = complete(x.right)
   if (h1 = h2 AND h1 ≠ -1)
        return h1 + 1
   else return -1
```

- Se x ha una foglia è basta è completo
- Se i due figli di x sono anche essi ~~compl~~ completi e hanno la stessa altezza
- Anche l'albero vuoto è completo

ES 1

Trovare nell'array il più piccolo indice tale che $A[i] > x$. Se l'indice esiste ritornarlo altrimenti ritorna $n+1$

$$\min \{ i \mid A[i] > x \}$$

```
over (A, p, r, x)
   if r = p + 1
      return p + 1
   else
      q = ⌊(p + r)/2⌋
      if A[q] > x
         return over(A, p, q, x)
      else
         return over(A, q+1, r, x)
```

→ siccome l'array è ordinato in modo crescente controllo l'elemento che sta nel mezzo e poi decido se ricorrere a SX o a DX

complessità

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

$a = 1$

$b = 2$

$n^{\log_b a} = n^0 = 1$, $f(n) = c \rightarrow \boxed{T(n) = \Theta(\log n)}$

ES 2

$$E = x_1 \; op_1 \; x_2 \; op_2 \; x_3 \; \cdots \; op_{n-1} \; x_n$$

$V_{i,j}$ = valore minimo di $x_i \, op_i \, x_{i+1} \cdots x_j$ $= \begin{cases} x_i & i = j \\ \min\{ V_{i,k} \; op_k \; V_{k+1,j} \mid 1 \leq k < j \} & j > i \end{cases}$

provo a mettere le parentesi in tutti i modi finché ottengo il minimo

$(x_i \; op_i \cdots x_k) \; op_k \; (x_{k+1} \cdots x_j)$

→ ParMin (X, op, n) // ritorna $V[i,j]$ = valore minimo della sottoespressione

for $i = 1$ to n
    $V[i,j] = x_i$
for $i = 1$ to n
    for $j = i+1$ to n
        $V[i,j] = -1$
        for $k = i$ to $j-1$
            $q = V[i,k] \; op_k \; V[k+1,j]$
            if $V[i,j] > q$
                $V[i,j] = q$
                $(p[i,j] = k)$
return $V[1,n]$, (p)

Complessità $\Theta(n^3)$

cose da aggiungere per avere la parentesizzazione

- Per stampare l'espressione devo sapere dove ho diviso (bisogna cambiare ParMin facendogli ritornare anche p) (4)

```
print (x, op, n)
    v, p = ParMin (x, op, n)
    printric (x, op, 1, n)


printric (x, op, i, j)
    if i = j
        echo xi
    else
        echo "("        — apro la parentesi
        K = p[i, j]
        printric (x, op, i, K)
        echo ")" . opK . "("
        printric (x, op, K+1, j)
        echo ")"
```