E 22/02/2023

martedì 7 marzo 2023 16:09

$$T(n) = \begin{cases} 1, & \text{se } n \le 27; \\ 3n^2 \cdot T(\sqrt[3]{n}) + 2n^3, & \text{altrimenti.} \end{cases}$$

LIVELLO		CONTRIBUTI SINGOLI	NUM RAMI	CONTRIBUTO TOTALE
0	$T(n) = 3n^2 \cdot T(\sqrt[3]{n}) + 2n^3$, $\downarrow T(n^{\frac{1}{3}})$	$2n^3$	1	$2n^3$
1	$T(n^{\frac{1}{3}}) = 3 \cdot (n^{\frac{1}{3}})^2 \cdot T((n^{\frac{1}{3}})^{\frac{1}{3}}) + 2 \cdot (n^{\frac{1}{3}})^3 = 3n^{\frac{2}{3}} \cdot T(n^{\frac{1}{9}}) + 2n$	$2n$	$3n^2$	$2n \cdot 3n^2$ $6n^3$
2	$T(n^{\frac{1}{9}}) = 3 \cdot n^{\frac{2}{9}} \cdot T((n^{\frac{1}{9}})^{\frac{1}{3}}) + 2 \cdot (n^{\frac{1}{9}})^3 = 3n^{\frac{2}{9}} \cdot T(n^{\frac{1}{27}}) + 2 \cdot (n^{\frac{1}{3}})$	$2n^{\frac{1}{3}}$	$3n^{\frac{2}{3}} \cdot 3n^2 = 9n^{\frac{8}{3}}$	$2n^{\frac{1}{3}} \cdot 9n^{\frac{8}{3}} = 18n^3$

CASO PEGGIORE

CONTRIBUTO GENERALE SOMMATORIA

$$2n^{\frac{3}{3^h}}$$

$$n^{\frac{1}{3^i}} \le 27 \qquad n^{\frac{1}{3^i}} = 27 \Rightarrow$$

$$\log_3 n^{\frac{1}{3^h}} = \log_3 27 \Rightarrow \frac{1}{3^h} \cdot \log_3(n) = 3 \Rightarrow$$

$$\log_3(n) = 3 \cdot 3^h \Rightarrow \log_3(n) = 3^{h+1} \Rightarrow$$

$$\Rightarrow h+1 = \log_3 \log_3(n) \Rightarrow h = \log_3(\log_3(n)) - 1$$

SOLUZIONE PERFETTA X MOGAVERO

$$n^{1/3^i} = 27$$

$$\log_{27} n^{1/3^i} = 1$$

$$\frac{1}{3^i} \cdot \log_{27}(n) = 1$$

$$\log_{27}(n) = 3^i \Rightarrow i = \log_3 \log_{27}(n)$$

$$T(n) = \sum_{i=0}^{h} 3^i \cdot 2n^3 = 2n^3 \sum_{i=0}^{h} 3^i = 2n^3 \cdot \left(\frac{3^{h+1} - 1}{3-1} \right)$$

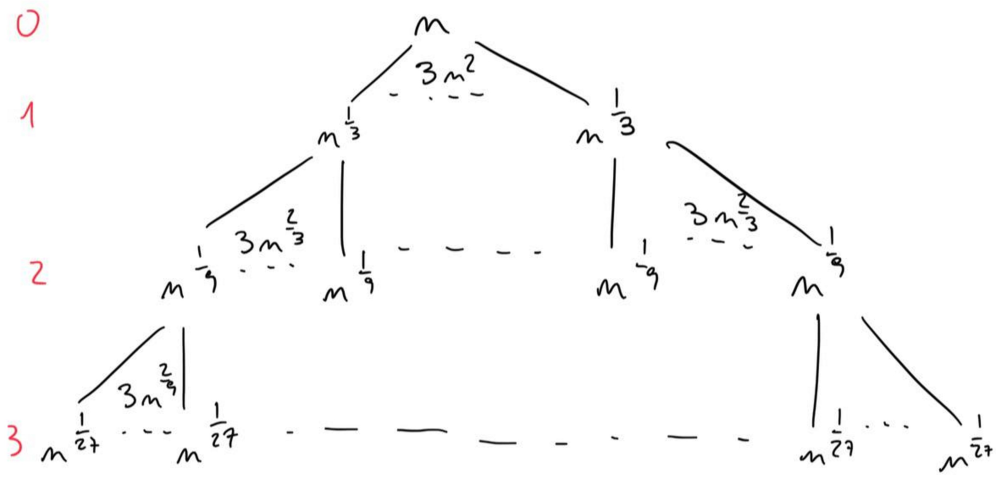
$$= \cancel{2} n^3 \cdot \frac{3^{h+1} - 1}{\cancel{2}} = n^3 \cdot \left(3^{\left(\log_3 (\log_3(n)) \cancel{-1} \cancel{+1} \right)} - 1 \right) =$$

$$n^3 \, 3^{\log_3(\log_3(n))} - n^3$$

$$\Theta\left(n^3 \cdot \log_3(n)\right)$$

SOLUZIONE LUCA

$$T(n) = \begin{cases} 1 & n \leq 27 \\ 3n^2 \cdot T(\sqrt[3]{n}) + 2n^3 & \text{altrimenti} \end{cases}$$



NUMERO NODI	LIVELLO
1	0
$3n^2$	1
$3n^2 \cdot 3n^{\frac{2}{3}}$	2
$3n^2 \cdot 3n^{\frac{2}{3}} \cdot 3n^{\frac{2}{9}}$	3

LIV.		CONTRIBUTO SINGOLO	CONTRIBUTO TOTALE
0	$3n^2 \cdot T(n^{\frac{1}{3}}) + 2n^3$	$2n^3$	$2n^3$
1	$3n^2\left(T(n^{\frac{1}{3}})\right) =$ $= 3n^2\left(3n^{\frac{2}{3}} \cdot T(n^{\frac{1}{9}}) + 2n^{\frac{3}{3}}\right)$	$2n^{\cancel{\frac{3}{3}}}$	$3n^2 \cdot 2n =$ $= 6n^3$
2	$3n^{\frac{2}{3}} \cdot 3n^2 \cdot \left(T(n^{\frac{1}{9}})\right) =$ $= \left(3n^{\frac{2}{9}} \cdot T(n^{\frac{1}{27}}) + 2n^{\cancel{\frac{3}{9}}\frac{1}{3}}\right)$ $\cdot 3n^{\frac{2}{3}} \cdot 3n^2$	$2n^{\cancel{\frac{3}{9}}\frac{1}{3}}$	$3n^2 \cdot 3n^{\frac{2}{3}} \cdot$ $2n^{\frac{1}{3}} =$ $= 18n^3$

POTREI ANDARE AVANTI, MA NON CREDO SIA NECESSARIO, IL CONTRIBUTO PER LIVELLO È:

$$3^i \cdot 2 \cdot n^3$$

MENTRE L'ALTEZZA DELL'ALBERO SEGUE LA REGOLA

$$n^{\left(\frac{1}{3}\right)^i}$$

E POSSO RICAVARE LA i COSÌ:

$$n^{\left(\frac{1}{3}\right)^i} = 27 \Rightarrow \log_{27} n^{\left(\frac{1}{3}\right)^i} = \cancel{\log_{27} 27}^{\,1}$$

$$\left(\frac{1}{3}\right)^i \log_{27} n = 1$$

$$\log_{27} n = 3^i$$

$$\log_3 \log_{27} n = i$$

OPPURE

$$n^{\left(\frac{1}{3}\right)^i} = 27 \Rightarrow \log_3 n^{\left(\frac{1}{3}\right)^i} = \cancel{\log_3 27}^{\,3}$$

$$\left(\frac{1}{3}\right)^i \log_3 n = 3 \Rightarrow \log_3 n = 3 \cdot 3^i$$

$$\log_3 n = 3^{i+1} \Rightarrow \log_3(\log_3 n) = \log_3\left(3^{i+1}\right)$$

$$\log_3(\log_3 n) = i+1 \left(\cancel{\log_3 3}^{\,1}\right)$$

$$i = \log_3(\log_3 n) - 1$$

CALCOLO DELLA SOMMATORIA

$$\sum_{i=0}^{h} 2 \cdot 3^i \cdot n^3 \Rightarrow 2n^3 \sum_{i=0}^{h} 3^i$$

$3 > 1$, USO LA FORMULA COMPLETA

$$2n^3 \left(\frac{3^{h+1} - 1}{3-1}\right) = \cancel{2} n^3 \cdot \frac{1}{\cancel{2}} \cdot \left(3^{h+1} - 1\right) =$$

$$= n^3 \cdot \left(3^{h+1} - 1\right)$$

SOSTITUISCO h COL RICAVATO DI PRIMA

$$= n^3 \cdot \left(3^{\log_3(\log_3 n) \cancel{-1} \cancel{+1}} - 1\right) =$$

$$= n^3 \cdot 3^{\log_3(\log_3 n)} - n^3 =$$

$$= n^3 \cdot 3^{\log_3(\log_3 n)} - n^3 =$$

$$= n^3 \cdot (\log_3 n)^{\cancel{\log_3 3}} - n^3 \qquad \boxed{\text{a^(log\_a (x)) = x}}$$

$$\Theta\left(n^3 \log_3(n)\right)$$

2

2. Si scriva un **algoritmo iterativo** che **simuli precisamente** l'algoritmo ricorsivo di seguito riportato, dove Z_l e Z_r sono due funzioni esterne non meglio specificate che soddisfano la seguente proprietà: $p < Z_l(A,p,s) < Z_r(A,p,s) \leq s$, quando $p+1 < s$.

```
function Algoritmo(A, p, s)
  if s ≤ p + 1 then
    return 0
  else
    q ← Z_l(A, p, s)
    r ← Z_r(A, p, s)
    a ← Algoritmo(A, p, q)
    a ← a − Algoritmo(A, q, r)
    a ← a + Algoritmo(A, r, s)
    return a + (r − q)
```

IF
C1
ELSE IF
C2
ELSE
C3

```
ALGO_IT(A,p,s)

    Cs = s
    Cp = p
    Start = true
    Last = NIL
    Ret = 0
    St_p = st_s = st_a = NIL


    While(start OR st_p != NIL) DO

        If (start) then

                If(cs <= cp + 1) THEN
                    Ret = 0
                    Start = false
                    Last = cp

                Else
                    // NUOVA CHIAMATA
                    /* SIMULO FINO ALLA 1 CHIAMATA */

                    q <- Zl(A,cp,cs)
                    r <- Zl(A,cp,cs)
                    St_p = push(st_p, cp)
                    St_s = push(st_s, cs)



        else   // VECCHIA CHIAMATA
                /* RECUPERO CONTESTO */
```

```
                Cp = top(st_p)
                Cs = top(st_s)
                q <- Zl(A,cp,cs)
                r <- Zl(A,cp,cs)


                If(last = cp) THEN //torno 1 CHIAMATA DA CHIAMARE LA 2

                    a <- ret
                    push(st_a, a)
                    Cp = q
                    Cs = r
                    Start = true

                else // 2 CHIAMATA

                    If(last = q) THEN   // torno dalla 2 CHIAMATA da CHIAMARE LA 3
                        a = top(st_a)
                        A <- a - ret
                        St_a = pop(st_a)
                        Push(st_a, a)
                        Cp = r
                        Cs = cs
                        Start = true



                    else    // TORNO DALLA 3 CHIAMATA ///// LAST = R

                        a = top(st_a)
                        a <- a + ret
                        St_a = pop(st_a)
                        Push(st_a, a)

                        Ret = a + (r - q)
                        St_p = pop(st_p)
                        St_s = pop(st_s)
                        Last = cp

                        Start = false



        Return ret

    }
```

```
AlgoritmoIter(A, p, s)
    cp = p
    cs = s
    st_p = st_s = st_a = NIL
    last = NIL
    start = true
    WHILE start || st_p != NIL DO
        IF start THEN
            IF cp + 1 > cs THEN
                q <- Z_l(A, cp, cs)
                r <- Z_r(A, cp, cs)
                st_p = push(st_p, cp)
                st_s = push(st_s, cs)
                cs = q
                /*start = true*/
            ELSE /*caso base cs <= cp + 1*/
                ret = 0
                last = cp
                start = false
        ELSE /*ripresa del contesto*/
            cp = top(st_p)
            cs = top(st_s)
            q <- Z_l(A, cp, cs)
            r <- Z_r(A, cp, cs)
            IF last = cp THEN
                a <- ret
                cp = q
                cs = r
                start = true
            ELSE IF last = q THEN
                a <- a - ret
                cp = r
                /*cs = cs*/
                start = true
            ELSE /* last = r*/
                a <- a + ret
                ret = a + (r - q)
                st_p = pop(st_p)
                st_s = pop(st_s)
                last = cp
                /*start = false*/
    RETURN ret
```

a <- ret
push(st_a, a)

a = top(st_a)
a <- a - ret
st_a = pop(st_a)
push(st_a, a)

a = top(st_a)
a <- a + ret
St_a = pop(st_a)

2023-02-22.jpg 110%

1. Si risolva la seguente equazione di ricorrenza, calcolandone l'**andamento asintotico**:

$$T(n) = \begin{cases} 1, & \text{se } n \leq 27; \\ 3n^2 \cdot T(\sqrt[3]{n}) + 2n^3, & \text{altrimenti.} \end{cases}$$

2. Si scriva un **algoritmo iterativo** che **simuli precisamente** l'algoritmo ricorsivo di seguito riportato, dove Z_l e Z_r sono due funzioni esterne non meglio specificate che soddisfano la seguente proprietà: $p < Z_l(A,p,s) < Z_r(A,p,s) \leq s$ quando $p + 1 < s$.

function Algoritmo(A, p, s)
  if $s \leq p + 1$ **then**
    return 0
  else
    $q \leftarrow Z_l(A, p, s)$
    $r \leftarrow Z_r(A, p, s)$
    $a \leftarrow$ Algoritmo(A, p, q)
    $a \leftarrow a -$ Algoritmo(A, q, r)
    $a \leftarrow a +$ Algoritmo(A, r, s)
    return $a + (r - q)$

ALGO_IT(A,p,s)

While(){

If(){ // NUOVA CHIAMATA
/* SIMULO FINO ALLA 1 CHIAMATA */

}else{ // VECCHIA CHIAMATA
/* RECUPERO CONTESTO */

If() { //torno 1 CHIAMATA con 2 CHIAMATA != da NULL

}else{ // 2 CHIAMATA

If(){ // torno dalla 1 CHIAMATA con 2 CHIAMATA = NULL

}else{ //TORNO DALLA 2 CHIAMATA
If(){ //torno dalla 2 CHIAMATA con 3 CHIAMATA != NULL

}else{ //TORNO dalla 3 CHIAMATA

}

}

```
    }

    Return

}
```