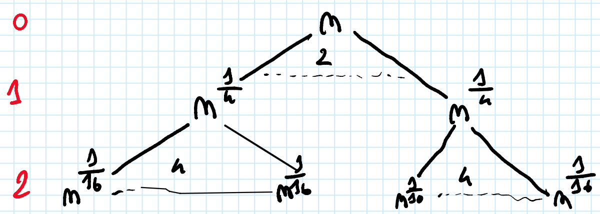
E 25/01/2023

martedì 7 marzo 2023 16:36

$$T(n) = \begin{cases} 1, & \text{se } n \le 2; \\ 2 \cdot T(\sqrt[4]{n}) + \log(2n), & \text{altrimenti.} \end{cases}$$

LIVELLO		CONTRIBUTI SINGOLI	NUM RAMI	CONTRIBUTO TOTALE
0	$T(n) = 2 \cdot T\left(n^{\frac{1}{4}}\right) + \log_2(2n)$	$\log(2n)$ $= 1 + \log(n)$	1	$\log(2n) =$ $= 1 + \log(n)$
1	$T\left(n^{\frac{1}{4}}\right) = 2 \cdot T\left(n^{\frac{1}{16}}\right) + \log\left(2\, n^{\frac{1}{4}}\right)$	$\log\left(2\, n^{\frac{1}{4}}\right)$ $= 2 + \frac{1}{4}\log(n)$	2	$2 \cdot \log\left(2\, n^{\frac{1}{4}}\right)$ $= 2 \cdot \left(\log(2) + \log\left(n^{\frac{1}{4}}\right)\right) =$ $= 2 + \frac{1}{2}\log(n)$
2	$T\left(n^{\frac{1}{16}}\right) = 2 \cdot T\left(n^{\frac{1}{64}}\right) + \log\left(2\, n^{\frac{1}{16}}\right)$	$\log\left(2\, n^{\frac{1}{16}}\right)$ $= 1 + \frac{1}{16}\log(n)$	$2 \cdot 2 = 4$	$4 \cdot \left(1 + \frac{1}{16}\log(n)\right)$ $= 4 + \frac{1}{4}\log(n)$

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



$$2^i + \frac{1}{2^i}\log(n)$$

MENTRE L'ALTEZZA DELL'ALBERO SEGUE LA REGOLA

$$n^{\left(\frac{1}{2^2}\right)^i}$$

RICAVIAMOCI L'ALTEZZA DELL'ALBERO $\boxed{H}$ SEGUENDO IL CASO PEGGIORE

$$n^{\frac{1}{2^{2h}}} = 2 \Rightarrow \log_2\left(n^{\left(\frac{1}{2^2}\right)^h}\right) = \log_2 2 \Rightarrow$$

$$\left(\frac{1}{2^2}\right)^h \log_2(n) = 1 \Rightarrow \frac{1}{2^{2h}}\log_2(n) = 1 \Rightarrow \log_2(n) = 2^{2h}$$

$$\log_2(\log_2(n)) = \log_2(2^{2h}) \Rightarrow \log_2(\log_2(n)) = 2h \Rightarrow h = \frac{\log_2(\log_2(n))}{2}$$

CALCOLO DELLA SOMMATORIA

$$\sum_{i=0}^{h} 2^i + \sum_{i=0}^{h} \frac{1}{2^i} \log(n) =$$

$$\sum_{i=0}^{h} 2^i = 2 > 1$$ QUINDI SERIE GEOMETRICA COMPLETA $$\Rightarrow \frac{2^{h+1} - 1}{2-1} = 2^{h+1} - 1$$

$$\Rightarrow 2^{\frac{\log_2(\log_2(n))}{2} + 1} - 1$$

$\log(n) \sum_{i=0}^{h} \frac{1}{2^n} = \frac{1}{2^h}$ E' COMPRESO TRA 0 E 1 $\Rightarrow \frac{1}{1 - \frac{1}{2}} = \frac{1}{\frac{1}{2}} = 2\log_2(n)$

$\boxed{\text{SFRUTTO LA PROPRIETA' } n = a^{\log_a(n)}}$

$$2^{\frac{\log_2(\log_2(n))}{2} + 1} - 1 + 2\log_2(n)$$

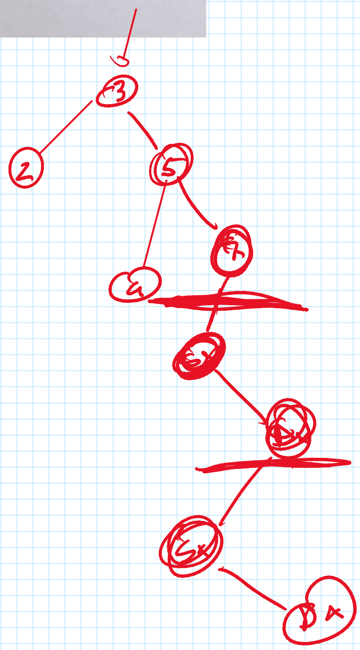
$$\sqrt{2^{\log_2(\log_2(n))}} \cdot 2 - 1 + 2\log_2(n) \Rightarrow \sqrt{\log_2(n)} \cdot 2 - 1 + 2\log_2(n)$$

$$\Theta(\log_2(n))$$

2

2. Si scriva un **algoritmo iterativo** che **simuli precisamente** l'algoritmo ricorsivo di seguito riportato, dove Z è una funzione esterna non meglio specificata.

```
function Algoritmo(T, h)
  if T = Nil then
    return Z(0, h)
  else
    a ← 0
    if T→key ≡ 0 mod 2 then
      a ← a + Algoritmo(T→dx, 2 · h)
    if T→key ≡ 1 mod 3 then
      a ← a − Algoritmo(T→sx, 3 · h)
    return Z(T→key, a)
```



```
ALGO_IT(T,h)

    st_t = st_h = st_a = NULL
    cH = h
    cT = T
    start = true
    last = NULL


    while(start = true OR st_t != NULL)

        if(start = true)
            if(cT = NIL)
                last = cT
                start = false
                ret = Z(0,cH)

            else

                st_t = push(st_t, cT)
                st_h = push(st_h, cH)

                if(cT->key = 0 mod 2)
                    cT = cT->dx
                    cH = 2 * cH
                else
                    start = false
                    ret = 0


        else
            cT = top(st_t)
            cH = top(st_h)

            if(cT->sx != last) // TORNO DA PRIMA CHIAMATA
                a = ret

                if(cT->key = 1 mod 3)
                    cT = cT->sx
                    cH = 3 * cH
                    start = true
                    st_a = push(st_a, a)
                else
                    st_t = pop(st_t)
                    st_h = pop(st_h)
                    start = false // RIPETITIVO
                    ret = Z(cT->key, a)
                    last = cT

            else // TORNO DA SECONDA CHIAMATA

                a = top(st_a)
                a <- a - ret
                start = false // RIPETITIVO
```

```
        ret = Z(cT->key, a)
        st_t = pop(st_t)
        st_h = pop(st_h)
        st_a = pop(st_a)
        last = cT


return ret
```

```
if j − i >= 1 then
    y = Rand()%2
    if (x = 1) then
        IF RET% 2-1 =0
        ret = ALGORITMO(A, (i+j)/2 + 1, j, y)
        if ret%2 = 0 then
            ret = ALGORITMO(A, i, (i+j)/2, 1 − y)
    else
        ret = ALGORITMO(A, i, (i+j)/2, y)
        if ret%2 = 1 then
            ret = ret+ALGORITMO(A, (i+j)/2 + 1, j, 1 − y)
else
    ret = A[i]
```