Estructuras de datos
2024-06-05
Análisis de algoritmos

¡Buenos días!

```
int recursiva (int n) {
    if (n <= 0)               cond
        return 43 ;           ret_1
    else
        return recursiva (n/3) ;   ret_2
}
```

1 1.1

$$T(n) = \begin{cases} n=0: T(cond) + T(ret_1) = t + t = 2t \\ n>0: T(cond) + T(ret_2) + T(n/3) = t + t + T(n/3) = 2t + T(n/3) \end{cases}$$

1.2 Expansion

$$T(n) = 2t + T(n/3) \qquad (1)$$

$$= 2t + [2t + T((n/3)/3)] = 2(2t) + T(n/3^2) \qquad (2)$$

$$= 2(2)t + [2t + T((n/3^2)/3)] = (3)2t + T(n/3^3) \qquad (3)$$

$$= (3)2t + [2t + T((n/3^3)/3)] = (4)2t + T(n/3^4) \qquad (4)$$

$$T(n) = K\,2t + T(n/3^K) \qquad (K)$$

$(2+2)t$ 4   6 $(2+2+2)$ 8 $\underbrace{(2+2+2+2)}_{K}$

7   2^3-1
15   2^4-1
31   2^5-1

1.3 Condición inicial: $T(\emptyset) = 2t$

$$\frac{n}{3^k} = \emptyset$$

$$T(1) = 2t + T(1/3) = 2t + T(\emptyset)$$

$$= 2t + 2t$$

$$= 4t$$

$$\frac{n}{3^k} = 1 \Rightarrow n = 3^k \Rightarrow \log_3(n) = K$$

$\Rightarrow$ Sust.

$$T(n) = \underline{K}\,2t + T\left(\frac{n}{3^k}\right) = 2t \cdot \log_3(n) + T\left(\frac{n}{n}\right)$$

$$= 2t \cdot \log_3(n) + T(1)$$

$$= 2t \cdot \log_3(n) + 4t$$

$$\Rightarrow T(n) = 2\log_3(n) + 4$$

2 Aplicar $O(n)$

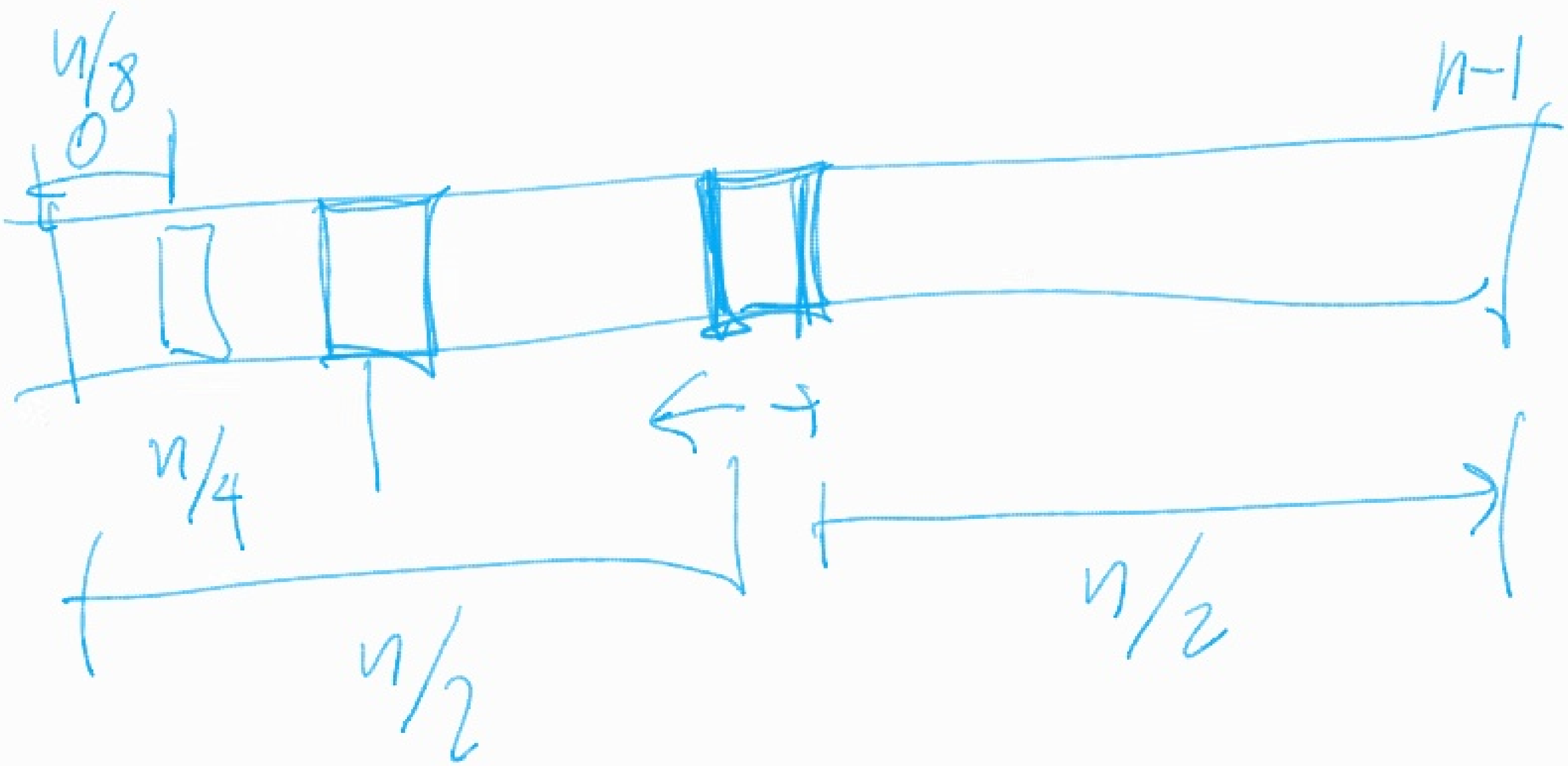
$$O(T(n)) = O(2\log_3(n) + 4)$$

R. suma

$$= \max(O(2\log_3(n)), O(4)) = O(2\log_3(n))$$

R. constantes

$$O(n) = \log_3(n)$$


n/8
0
n-1
n/4
n/2
n/2

Pr-1.3 - optimización

```
int recursiva (int n) {

    if (n <= 1)            cond

        return 23 ;        ret1
    else

        return recursiva (n/2) + recursiva (n/2) ;        ret2
}
```

$$T(n) = \begin{cases} n=1 : T(cond) + T(ret_1) = t + t = \underline{2t} \\ n>1 : T(cond) + T(ret_2) + T(n/2) + T(n/2) \\ \quad = t + t + 2T(n/2) = 2t + 2T(n/2) \end{cases}$$

K-ésima exp.

$$T(n) = (2^1 + 2^2 + \dots + 2^K)t + 2^K \, T(n/2^K)$$

(1)

$$T(n) = 2t + 2\underline{T(n/2)}$$

$$= 2t + 2\left[2t + 2T\left(\frac{n/2}{2}\right)\right] = (2 + 2(2))t + 2^2\,T(n/2^2) = (2^1 + \underline{2^2})t + 2^2\,\underline{T(n/2^2)}$$

(2)

$$= (2^1 + 2^2)t + 2^2\left[2t + 2T\left(\frac{n/2^2}{2}\right)\right] = (2^1 + 2^2 + \underline{2^3})t + 2^3\,T(n/2^3)$$

(3)

$$T(n) = (2^1 + 2^2 + \dots + 2^k)t + 2^k T(n/2^k)$$

si $T(1) = 2t$

$\Rightarrow n/2^k = 1 \Rightarrow n = 2^k$

$\Rightarrow k = \lg(n)$

sust. 2^k

$$T(n) = (2^1 + 2^2 + \dots + n)t + nT(n/n)$$

$$= (2^1 + 2^2 + \dots + n)t + nT(1)$$

$$= 2^1 + 2^2 + \dots + n + 2n$$

$$O(T(n)) = O(2^1 + 2^2 + 2^3 + \dots + n + 2n)$$

R. suma

$$= \max(O(2^1), O(2^2), \dots, O(n), O(2n))$$

$$= O(2n)$$

R. constantes

$$O(n) = n \,//$$

```
int recursive (int n) {

    if (n <= 1)

        return 23 ;
    else

        return recursiva (n/2) + recursiva (n/2) ; *2
}
```

$$\begin{array}{ccc} T(n) & & O(T(n)) \\ < & \Rightarrow & < \\ T(2n) & & O(T(2n)) \end{array}$$

$$T(n)+T(2n) < T(2n)+T(2n)$$

$$2T(2n)$$