Tiempo de ej

viernes, 12 de mayo de 2023 14:08

$\boxed{\text{Tiempo iterativo}}$

$$\sum_{i=1}^{N} c = c.N$$ → Algo q' no depende de i

$$\sum_{i=0}^{N} c \qquad N+1 \cdot c \qquad Nc + c$$

$$\sum_{i=A}^{B} \quad (B-A+1) \cdot c$$ → si no arranca en 1, ajusto

cuando lo de la Σ depende del índice

$$\sum_{i=1}^{N} i = 1+2+3 \ldots + N-2 + N-1 + N$$ } si uno suma a izquierda de extremos, n+1 siempre

$$\sum_{i=1}^{N} i = (n+1) \cdot \frac{n}{2} = \boxed{\frac{n(n+1)}{2}}$$ si es la Σ del índice

$$\sum_{i=a}^{N} i = \sum_{i=1}^{n} - \sum_{i=1}^{a-1}$$ si no arranca en 1

$$\sum_{i=1}^{n-1}$$

$$\frac{n(n+1)}{2} - \frac{(a-1).(a-1+1)}{2} = \frac{n^2+n}{2} - \frac{a^2-a}{2} = \frac{1}{2}n^2 + \frac{n}{2} - \frac{a^2}{2} + \frac{a}{2}$$

- for i=1 to N, i++ → $T(n) = \sum_{i=1}^{N} c$
  Algo
- for i=n to 1, i++ → $T(n) = \sum_{i=1}^{N} c$
  algo
- for i=1 to N, i+2 → $T(n) = \sum_{i=1}^{N/2} c$ } se hace la mitad de las veces el for
  algo
- c=1
  while c<N → $T(n) = \sum_{i=1}^{\log_2(N)} c$
  c=c.2

paso	c
1	$1 = 2^0$
2	$2 = 2^1$
3	$4 = 2^2$
4	$8 = 2^3$
i	2^{i-1}

$c = N$ → último paso

$2^{i-1} = N$

$i-1 = \log_2(n)$

$i = \log_2(n) + 1$ — NO + 1 pq ahí () cuando falla.

- For i=1 to N
  - FOR j=i to N
    - FOR k=1 to N
      - c

$$\sum_{i=1}^{N}\left[\sum_{j=i}^{N}\left[\sum_{k=1}^{N} c\right]\right]$$

$$\sum_{i=1}^{N}\left[\sum_{j=i}^{N} cN\right]$$

$$\sum_{i=1}^{N}(N-i+1) \cdot cN \overset{A}{=} \sum_{i=1}^{N}\left(cN^2 - cNi + Nc\right) \overset{A}{=} \sum_{i=1}^{N} cN^2 + \sum_{i=1}^{N} -cNi + \sum_{i=1}^{N} Nc =$$

$$\overset{B}{=} cN \sum_{i=1}^{N} N-i+1$$

$$\overset{B}{=} \sum_{i=1}^{N} cN^2 + Nc + \sum_{i=1}^{N} -cNi = (cN^2+Nc).N - cN\sum_{i=1}^{N} i = cN^3 + N^2c - cN.\frac{n.(n+1)}{2}$$

$$cN^3 + N^2c - \frac{cN^2(n+1)}{2}$$

$$cN^3 + N^2c - \frac{cN^3+cN^2}{2}$$

$$cN^3 + cN^2 - \frac{cN^3}{2} - \frac{cN^2}{2}$$

$$\frac{c}{2}N^3 + \frac{c}{2}N^2$$

- FOR i=1 to N
  - For j=i to N
    - c
    - FOR k=i to j
      - c2

$$\sum_{i=1}^{N}\left(\sum_{j=i}^{N}\left(c+\sum_{k=i}^{j} D\right)\right)$$

$$\sum_{i=1}^{N}\left(\sum_{j=i}^{N}\left(c+(j-i+1).D\right)\right)$$

$$\sum_{i=1}^{N}\left(\sum_{j=i}^{N} c + \sum_{j=i}^{N} D.(j-i+1)\right)$$

$$\sum_{i=1}^{N}\left(c(N-i+1) + D\sum_{j=i}^{N} j + \sum_{j=i}^{N} -i+1\right)$$

$$cN - ci + c + D\sum_{j=1}^{N} j - \sum_{j=1}^{i-1} j +$$

$\boxed{\text{Tiempo recursivo}}$

- factorial(n)
  - if (n=1)
    - return(1)
  - else
    - return (n * factorial(n-1))

$$T(n) = \begin{cases} c & N=1 \\ c + T(N-1) & N>1 \end{cases}$$

$p_1 \quad T(N) = C + T(N-1) \qquad N>1$

$p_2 \quad = C + C + T(N-1-1) \qquad N-1>1$

$p_3 \quad = C + C + C + T(N-1-1-1) \qquad N-2>1$

$N+1 =$

$S = N-1$

$p_S \quad = CS + T(N-S) \rightarrow$ paso S

De dónde sale esto?

$C \cdot N-1 + T(N-(N-1))$

$C(N-1) + T(1)$

$C(N-1) + D$

$$T(N) = \begin{cases} C & N=1 \\ (N-1) \cdot C + D & N>1 \end{cases}$$

- Escribir recurrencia
- Desarrollar recurrencia hasta que tengo que ir cambiando?
- Igualar argumento a caso base → lo que va cambiando. El resto queda igual. Tiene que quedar T(1)

$$T(n) = \begin{cases} C & N=1 \\ 2T(N/2) + N & N>1 \end{cases}$$

Da la recurrencia

paso 1 $\quad T(N) = 2T(N/2) + N \qquad N>1$

Acá N es (N/2)

paso 2 $\quad 2\left(2T\left(\frac{N/2}{2}\right) + \frac{N}{2}\right) + N \longrightarrow 2^2\left(\frac{N}{4}\right) + \frac{N}{2} + N$

paso 3 $\quad 2^2\left(2T\left(\frac{N}{8}\right) + \frac{N}{4}\right) + 2N$

$S \quad 2^S\, T\left(\frac{N}{2^S}\right) + SN$

$\frac{N}{2^S} = 1$

$N = 2^S$

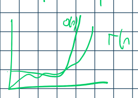
$S = \log_2 N$

reemplazo

$2^{\log_2 N}\, T\left(\frac{N}{2^{\log_2(N)}}\right) + \log_2(N) \cdot N$

$T(N)$ es $O(N)$ si y solo si

$T(N) \leq c\, O(N)$?



$N^2 + 100 N \quad O(N^2)$

$N^2 + 100 \leq N^2 \qquad \forall N \geq N_0$

$N^2 \leq C_1 N^2 \qquad \forall N \geq N_{01}$

$C_1 = 1$

$N_{01} = 1$

$100 \leq C_2 N^2 \qquad \forall N \geq N_{02}$

$C_2 = 100$

$N_{02} = 1$

```
Private int recu(int x, int n){
   int answer=0;
   if(n==1){
     answer=x;
   }else{
     answer=recu(x,n/2);
     answer+=recu(x,n/2);
     for(int j=1;j<=n;j++)
        answer+=x;
   }
   return answer;
}
```

C

D

$$T(n) = \begin{cases} C & N=1 \\ D + 2T\left(\frac{n}{2}\right) + \sum_{j=1}^{n} E & N>1 \end{cases}$$

$D + 2T\left(\frac{n}{2}\right) + n \cdot E$

$1 \rightarrow T(N) = D + 2T\left(\frac{N}{2}\right) + NE \qquad N>1$

$2 \rightarrow T\left(\frac{N}{2}\right) = 3D + 4T\left(\frac{N}{4}\right) + 2E\frac{N}{2} \qquad N/2 > 1$

$\rightarrow D + 2\left(D + 2T\left(\frac{N}{4}\right) + E\frac{N}{2}\right) + EN$

$3 \rightarrow T\left(\frac{N}{4}\right) = 3D + 4\left[D + 2T\left(\frac{N}{8}\right) + E\frac{N}{4}\right] + 2EN$

$7D + 8T\left(\frac{N}{8}\right) + 3EN$

$(2^3 - 1)D + 2^3 T\left(\frac{N}{2^3}\right) + 3EN$

$i \rightarrow (2^i - 1)D + 2^i T\left(\frac{N}{2^i}\right) + iEN$

$\rightarrow (2^{\log_2(n)} - 1)D + 2^{\log_2(n)}\left(\frac{N}{2^{\log_2 n}}\right) + \log_2(n)$

$(N-1)D + N + \left(\frac{N}{N}\right) + \log_2(N)\, EN$

$(N-1) \cdot D + CN + \log_2(N)\, EN \rightarrow O(N \log(n))$

$\frac{N}{2^i} = 1$

$N = 2^i$

$i = \log_2(n)$