1.1 INTRODUCCIÓN AL ANÁLISIS DE ALGORITMOS

¿Qué analizar?
- corrección
- uso de memoria
- rendimiento (tiempo de ejecución)

¿cómo medir el rendimiento?
↳ analizando el pseudocódigo: tiempo abstracto de ejecución (TAE)

¿CÓMO MEDIR EL TAE?

Ejemplo: multiplicar matrices (MM)

$$tae_{MM}(A,B,N) = \sum_{i=1}^{N} \underbrace{tae_i(A,B,N)}_{\sum_{j=1}^{N} \underbrace{tae_{ij}(A,B,N)}_{\sum_{k=1}^{N} tae_{ijk}(A,B,N)} + 1}$$

$$tae_{ijk}(A,B,N) = 1 \Rightarrow \sum_{k=1}^{N} tae_{ijk}(A,B,N) = \sum_{k=1}^{N} 1 = N$$

$$tae_{ij}(A,B,N) = N \Rightarrow \sum_{j=1}^{N} tae_{ij}(A,B,N) + 1 = \sum_{j=1}^{N} (N+1) = N(N+1)$$

$$tae_i(A,B,N) = N(N+1) \Rightarrow \sum_{i=1}^{N} tae_i(A,B,N) = \sum_{i=1}^{N} N(N+1) = N(N(N+1))$$

Le sumamos el 1 del return C.

$$\Rightarrow tae_{MM}(A,B,N) = N(N(N+1)) + 1 = N^3 + N^2 + 1$$

Análisis SelectSort

$$n_{ss}(T,1,N) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} 1 = \sum_{i=1}^{N-1} (N-i) =$$

$$= \sum_{i=1}^{N-1} N - \sum_{i=1}^{N-1} i = N(N-1) - \frac{N(N-1)}{2} = \frac{N(N-1)}{2} =$$

$$= \frac{N^2}{2} - \frac{N}{2}$$

Análisis Búsqueda Binaria

$$n_{BB}(T,1,N,k) = \cancel{\sum_{?}^{?}} 1$$

interacción $\underline{1}$: $\left\lceil \frac{N-1}{2} \right\rceil \leq \frac{N}{2^1}$

interacción $\underline{2}$: $\leq \frac{N}{2^2}$

$\vdots$

interacción $\underline{k}$: $\leq \frac{N}{2^k}$

$$\frac{N}{2^k} \leq 1 < \frac{N}{2^{k-1}} \xRightarrow{\text{dividimos por } N} 2^{k-1} < N \leq 2^k \Rightarrow$$

$$\Rightarrow \log_2 2^{k-1} < \log_2 N \leq \log_2 2^k \Rightarrow k-1 < \underset{\lg N}{\underset{\shortparallel}{\log_2 N}} \leq k$$

$$n_{BB}(T,1,N,k) \leq \lceil \log_2 N \rceil$$

Análisis de ordenación por burbuja

$$n_{BS}(T,1,N) = \sum_{i=N}^{2} \text{algo más} = \sum_{2}^{i=N} \sum_{j=1}^{i-1} 1 = \sum_{2}^{i=N} (i-1) =$$

nos da igual sumar del último al primero que del primero al último

$k = i-1$

$$= \sum_{1}^{N-1} k = \frac{(N-1)N}{2} = \frac{N^2}{2} - \frac{N}{2}$$

COMPARACIÓN DE ALGORITMOS

- Método a seguir

1. $\forall$algoritmo $A \in F$ encontrar $f_A(N)$ tal que $n_A(I) \leq f_A(\tau(I))$ con $\tau(I)$ el tamaño de entrada.

2. Diremos que A_1 es mejor que A_2 si:
$f_{A_1}(N)$ es "menor" que $f_{A_2}(N)$

- Sólo tiene sentido la comparación para entradas "grandes" = asintótica
- Sólo compararemos $f_{A_1}(N)$ y $f_{A_2}(N)$ cuando N es grande (es decir para $N \longrightarrow \infty$).

1.2 ESTIMACIÓN DEL CRECIMIENTO DE FUNCIONES

DEFINICIÓN 1 ($f = o(g)$, f "<" g): Si f y g son funciones positivas, decimos que $f = o(g)$ si $\boxed{\lim_{N\to\infty} \frac{f(N)}{g(N)} = 0}$

Ejemplos:

① $f = N^K$, $g = N^{K+\varepsilon}$

$$\frac{f}{g} = \frac{N^K}{N^{K+\varepsilon}} = \frac{1}{N^{\varepsilon}}_{\forall \varepsilon>0} \xrightarrow{N\to\infty} 0$$

② $f = \log(N)$, $g = N^{\varepsilon}$

$$\frac{f}{g} = \frac{\log(N)}{N^{\varepsilon}} \xrightarrow{N\to\infty} \frac{\infty}{\infty} \text{ IND}$$

$$\frac{f'}{g'} = \frac{1/N}{\varepsilon N^{\varepsilon-1}} = \frac{1}{\varepsilon N^{\varepsilon}} \xrightarrow{N\to\infty} 0$$

DEFINICIÓN 2 ($f = O(g)$, f "$\leq$" g): $f = O(g)$ si existen N_0 y $C > 0$ tales que para todo $n \geq N_0$, $f(N) \leq C g(N)$

Ejemplo

$f = N^2$; $g = N^2 + \sqrt{N}$

y también $g = O(f)$

Interpretación: g es mayor o igual que f eventualmente y con ayuda

↳ "demostración" (con la definición de límite)

$\forall \varepsilon (= 2$ aquí$)$ $\exists N_\varepsilon$ tal que $\frac{f(N)}{g(N)} \leq \varepsilon (= 2$ aquí$)$ si $n \geq N_\varepsilon$

$$\Rightarrow f(N) \leq \underset{\substack{\downarrow \\ \varepsilon \equiv C}}{2}\, g(N) \quad \forall n \geq N_\varepsilon$$

observaciones sobre $f = O(g)$

- $f = o(g) \Rightarrow f = O(g)$ pero <u>NO</u> $f = O(g) \not\Rightarrow f = o(g)$ (recíproco)
- Las constantes "no importan" en O, es decir, si:
  $\underline{f = O(g) \Rightarrow f = O(kg)}$, donde $k > 0$
- Si $f = O(g)$ y $h = o(g)$ entonces $f = O(g+h)$ pues $g + h = O(g)$
  - Decir $f = O(N^2 + N)$ no añade precisión, basta con $\underline{f = O(N^2)}$

<u>DEFINICIÓN 3</u> ($f = \Omega(g)$, f "$\geq$" g)

$$f = \Omega(g) \Leftrightarrow g = O(f)$$

<u>DEFINICIÓN 4</u> ($f = \Theta(g)$, f "$=$" g)

$$f = \Theta(g) \text{ si } f = O(g) \text{ y } g = O(f)$$

es fácil observar que: $f = \Theta(g) \Leftrightarrow g = \Theta(f)$

Si $\boxed{\lim_{N \to \infty} \frac{f(N)}{g(N)} = L \neq 0 \Rightarrow f = \Theta(g)}$

<u>DEFINICIÓN 5</u> ($f \sim g$): Decimos que f es asintóticamente equivalente a g si $\boxed{\lim_{N \to \infty} \frac{f(N)}{g(N)} = 1}$

<u>Ejemplo</u>

$f(N) = N^2 + \sqrt{N} + \log(N)$; $g(N) = N^2$

<u>observación</u>

$f \sim g \Rightarrow f = \Theta(g)$ pero <u>NO</u> $f = \Theta(g) \not\Rightarrow f \sim g$ (recíproco)

DEFINICIÓN 6 $(f = g + O(h))$: Si $h = o(g)$ entonces decimos que $f = g + O(h)$ si $|f - g| = O(h)$

Ejemplo

$$f(N) = N^2 + \sqrt{N} + \log(N) \quad ; \quad g(N) = N^2$$

Entonces $f = g + O(\underbrace{\sqrt{N}}_{h})$

ESCALA DE PRECISIÓN DECRECIENTE

- Si $f = g + O(h)$ entonces $f \sim g$
- Si $f \sim g$ entonces $f = \Theta(g)$
- Si $f = \Theta(g)$ entonces $f = O(g)$

Pero los recíprocos son falsos

Además:

- Si $f = O(g)$ y $f' = O(g')$ entonces

$$f + f' = O(g + g')$$
$$f \cdot f' = O(f \cdot f')$$

- CONSECUENCIA: si $P(N)$ es un polinomio de grado K se tiene que: $P(N) = a_K N^K + O(N^{K-1})$

CRECIMIENTO DE PROGRESIONES ARITMÉTICAS Y GEOMÉTRICAS

$$f(N) = S_N = \sum_{i=1}^{N} i = \frac{N(N+1)}{2} = \frac{N^2}{2} + O(N)$$

PROGRESIÓN GEOMÉTRICA (muy importante)

$$S_N = \sum_{i=1}^{N} x^i = \frac{x^{N+1} - x}{x-1} = \frac{UR - P}{R-1}$$

Observaciones:

- Si $x = 1$ $\quad S_N = N$
- Si $x > 1$ $\quad S_N = \Theta(x^N)$
- Si $x < 1$ $\quad S_N = \frac{x - x^{N+1}}{1-x} \xrightarrow{N \to \infty} \frac{x}{1-x}$

CRECIMIENTO DE FUNCIONES DERIVADAS

- Serie derivada $S_N = \sum_{i=1}^{N} i.x^i = x \cdot \frac{d}{dx} \sum_{i=1}^{N} x^i = \Theta(Nx^N)$

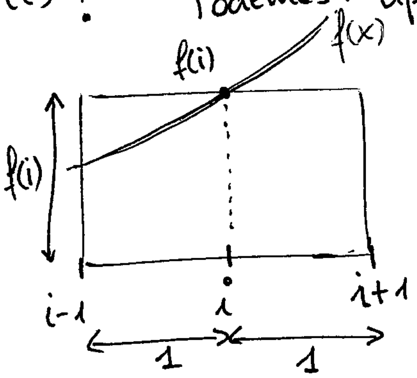
- Suma de potencias cúbicas:

$$S_N = \sum_{i=1}^{N} i^2 = \frac{N(N+1)(2N+1)}{6} = \frac{N^3}{3} + O(N^2)$$

- Nos interesa el crecimiento y no tanto la fórmula cerrada
- Se puede simplificar?

ESTIMACIONES DE CRECIMIENTO DE SUMAS

¿Qué hacer cuando no tenemos una expresión cerrada para una suma $S_N = \sum_{i=1}^{N} f(i)$? Podemos aproximar la suma mediante <u>integrales</u>



$$\int_{i-1}^{i} f(x)\,dx \le f(i) \le \int_{i}^{i+1} f(x)\,dx \Rightarrow \sum_{i=1}^{N}\int_{i-1}^{i} f(x)\,dx \le \sum_{i=1}^{N} f(i) \le \sum_{i=1}^{N}\int_{i}^{i+1} f(x)\,dx \Rightarrow$$

$$\Rightarrow \boxed{\int_0^N f(x)\,dx \le S_N \le \int_1^{N+1} f(x)\,dx}$$ $f(x)$ tiene que ser CRECIENTE

<u>Ejemplos</u>

1. $S_N = \sum_{1}^{N} i^K$; $f(x) = x^K$ (creciente)

$$\int_0^N f(x)dx \le S_N \le \int_1^{N+1} x^K dx = \left[\frac{x^{K+1}}{K+1}\right]_1^{N+1} = \frac{(N+1)^{K+1}}{K+1} - \frac{1}{K+1}$$

$$\parallel$$

$$\left[\frac{x^{K+1}}{K+1}\right]_0^N = \frac{N^{K+1}}{K+1}$$

Entonces: $\frac{N^{K+1}}{K+1} \le S_N \le \frac{(N+1)^{K+1}}{K+1} - \frac{1}{K+1} \Rightarrow$ dividimos todo por $\frac{N^{K+1}}{K+1}$

$$\frac{\frac{N^{K+1}}{K+1}}{\frac{N^{K+1}}{K+1}} \le \frac{S_N}{\frac{N^{K+1}}{K+1}} \le \frac{(N+1)^{K+1}}{\cancel{K+1}} \cdot \frac{\cancel{K+1}}{N^{K+1}} - \frac{1}{\cancel{K+1}} \cdot \frac{\cancel{K+1}}{N^{K+1}} \Rightarrow$$

$$\Rightarrow 1 \le \frac{S_N}{\frac{N^{K+1}}{K+1}} \le \overbrace{\left(\frac{N+1}{N}\right)^{K+1} - \frac{1}{N^{K+1}}}$$

cuando $n \to \infty$ esto $:= 1 - 0 = 1$

$$\Rightarrow 1 \le \frac{S_N}{N^{K+1}/K+1} \le 1 \Rightarrow$$

$$\Rightarrow S_N \sim \frac{N^{K+1}}{K+1}$$

Vamos a intentar poner $S_N = \frac{N^{K+1}}{K+1} + O(\text{algo})$

Ahora bien, es fácil ver que $S_N = \frac{N^{K+1}}{K+1} + O(N^K)$

Demostración: $¿ \left| S_N - \frac{N^{K+1}}{K+1} \right| \overset{?}{=} O(N^K)$?

como vimos que $\frac{N^{K+1}}{K+1} \leq S_N$ podemos quitar los valores absolu

$$S_N - \frac{N^{K+1}}{K+1} \leq \frac{(N+1)^{K+1}}{K+1} - \frac{N^{K+1}}{K+1} - \frac{1}{K+1} = \frac{1}{K+1}\left((N+1)^{K+1} - N^{K+1} - 1\right) =$$

$$= \underbrace{\cancel{N^{K+1}} + aN^K + bN^{K-1} + \dots}_{\text{binomio de Newton}} - \cancel{N^{K+1}} - 1 \Rightarrow$$

$$\Rightarrow S_N - \frac{N^{K+1}}{K+1} \leq N^K \Rightarrow S_N = \frac{N^{K+1}}{K+1} + O(N^K) \quad \text{demostrado}$$

---

(2) $S_N = \sum_1^n \log i = \log 1 + \log 2 + \dots + \log N = \log(1.2.\dots N) = \log(N!)$

$$\int_0^N \log x\, dx \leq S_N \leq \int_1^{N+1} \log x\, dx \Rightarrow \text{por partes} \Rightarrow$$

$$\left[x\log x - x\right]_0^N \leq S_N \leq \left[x \log x - x\right]_1^{N+1} \Rightarrow$$

tiende a cero cuando $x \to \infty$ (L'H')

$$\Rightarrow N\log N - N - 0.\log 0 + 0 = N\log N - N \leq S_N \leq (N+1)\log(N+1) - (N+1) + 1$$

$$= N\log(N+1) + \log(N+1) - N \Rightarrow \underline{N\log N} - N \leq S_N \leq N\log(N+1) + \log(N+1) -$$

dividimos todo por $N\log N$:

$$\underset{\substack{\downarrow \\ 1}}{1} - \frac{1}{\underset{\substack{\downarrow n\to\infty \\ 0}}{\log N}} \leq \frac{S_N}{N\log N} \leq \underbrace{\frac{\log(N+1)}{\log N}}_{\to 1} + \underbrace{\frac{\log(N+1)}{\log N}}_{\to 1\ (L'H)} \cdot \underset{\to 0}{\frac{1}{N}} - \frac{1}{\log N} \xrightarrow{n\to\infty} 0 \Rightarrow$$

$$\Rightarrow 1 \leq \frac{S_N}{N\log N} \leq 1 \text{ cuando } n \to \infty \Rightarrow \boxed{S_N \sim N\log N}$$

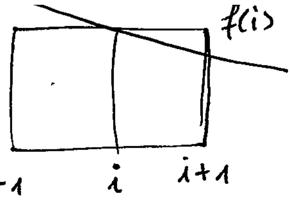
$¿S_N = N\log N + O(N)$? $\to |S_N - N\log N|$

(*) $\log N! \sim N\log N$

$N! \sim e^{N\log N}$

fórmula Stirling

③ $H_N = \sum_1^N \frac{1}{i}$; $f(x) = \frac{1}{x}$



esto es decreciente

$$\Rightarrow \int_0^N f(x)\,dx \geq H_N \geq \int_1^{N+1} f(x)\,dx$$

$$\Rightarrow [\log x]_0^N \geq H_N \geq [\log x]_1^{n+1} \Rightarrow$$

$$\Rightarrow \underbrace{\log N + \infty}_{\text{desigualdad inútil}} \geq H_N \geq \dots$$

Lo arreglamos así: $H_N = 1 + \sum_2^N \frac{1}{i}$; $f(x) = 1/x$

$$1 + \int_1^N f(x)\,dx \geq H_N \geq 1 + \int_2^{N+1} f(x)\,dx$$

$$1 + \int_1^N \frac{dx}{x} \geq H_N = 1 + \sum_2^N \frac{1}{i} = 1 + S_N \geq 1 + \int_2^{N+1} \frac{dx}{x}$$

$$1 + [\log x]_1^N = 1 + \log N \geq 1 + S_N \geq 1 + \log(N+1) - \log 2$$

$$\boxed{\log N \geq S_N \geq \log(N+1) - \log 2}$$

dividimos por $\log N$

$$1 \geq \frac{S_N}{\log N} \geq \underbrace{\frac{\log N+1}{\log N}}_{\substack{\downarrow N\to\infty \\ L'H \\ 1}} - \underbrace{\frac{\log 2}{\log N}}_{\substack{\downarrow \\ 0}} \Longrightarrow S_N \sim \log N$$

$$\boxed{H_N = 1 + S_N \sim 1 + \log N \sim \log N}$$

Ahora bien, ¿ $H_N = \log N + O(?)$?

Habría que apostar por ¿ $H_N = \log N + O(1)$? (muy complicado)

NOTA:

$$H_N - \log N \xrightarrow[N\to\infty]{} \gamma \text{ (constante de Euler Mascheroni)}$$

1.3 COMPLEJIDAD DE ALGORITMOS

Hasta ahora, el trabajo realizado por un algoritmo dependía de cada entrada en particular: $n_A(I) \leq f_A(\tau(I))$

Nos gustaría precisar la definición del trabajo que realiza un algoritmo

Para ello definimos el ESPACIO DE ENTRADAS de tamaño N de un algoritmo A como:

$$E_A(N) = \{ I \text{ entrada de } A \,/\, \tau(I) = N \}$$

CASOS MEJOR, PEOR, MEDIO

CASO PEOR: $W_A(N) = \max\{ n_A(I) \,/\, I \in E_A(N) \}$

CASO MEJOR: $B_A(N) = \min\{ n_A(I) \,/\, I \in E_A(N) \}$

CASO MEDIO: $A_A(N) = \sum_{I \in E_A(N)} n_A(I) \cdot p(I)$

donde $p(I)$ es la probabilidad con que aparece la entrada I.

¿cómo estimar el caso peor de un algoritmo?

① Encontrar $f_A(N)$ tal que si $\tau(I) = N$ se tiene:

$$n_A(I) \leq f_A(N)$$

② Encontrar una entrada $\hat{I}$, que sea la entrada peor del algoritmo tal que: $n_A(\hat{I}) \geq f_A(N)$

① $W_A(N) \leq f_A(N)$
② $W_A(N) \geq f_A(N)$

¿cómo estimar el caso mejor de un algoritmo?

① Encontrar $f_A(N)$ tal que si $\tau(I) = N$ se tiene:

$$n_A(I) \geq f_A(N)$$

② Encontrar una entrada $\hat{I}$, que sea la entrada mejor del algoritmo tal que: $n_A(\hat{I}) \leq f_A(N)$

① $B_A(N) \geq f_A(N)$
② $B_A(N) \leq f_A(N)$

- CASO MEDIO Búsqueda Lineal

$$K \begin{cases} \text{éxito} \Leftrightarrow K \in T \\ \text{fracaso} \Leftrightarrow K \notin T \end{cases}$$

$$E_{BL}(N) = \underbrace{\{1 \text{ — } N\}}_{\text{éxito}} \cup \underbrace{\{\text{otra}\}}_{\text{fracaso}} \longleftrightarrow |E_{BL}(N)| = N+1$$

Equiprobabilidad $\rightarrow p(k==i) = \frac{1}{N+1}$

$\rightarrow p(k==\text{otra}) = \frac{1}{N+1}$

$$A_{BL}(N) = \underbrace{\sum_1^N n_i \cdot p_i}_{\text{éxito}} + \underbrace{\overset{\text{comparaciones}}{N} \cdot \overset{\text{probabilidad}}{\frac{1}{N+1}}}_{\text{fracaso}} = \underbrace{\sum_{i=1}^{N} i \frac{1}{N+1}}_{\text{éxito}} + \underbrace{N \cdot \frac{1}{N+1}}_{\text{fracaso}} =$$

$$= \frac{1}{N+1} \cdot \frac{N(N+1)}{2} + \frac{N}{N+1} = \frac{N}{2} + \frac{N}{N+1} = \frac{N}{2} + O(1)$$

$$\boxed{W_{BL}(N) = N} \quad \boxed{B_{BL}(N) = 1} \quad \boxed{A_{BL}(N) = \frac{N}{2} + O(1)}$$

$$p(K == T[i]) = p(\text{hacer } i \text{ comparaciones}) = \frac{1}{C_N} f(i)$$

Como $1 = \sum_1^N \frac{1}{C_N} f(i) \Rightarrow C_N = \sum_1^N f(i)$

2.1 ALGORITMOS DE ORDENACIÓN

INSERTSORT

```
InsertSort(Tabla T, ind P, ind U)
    para i de P+1 a U;
        A = T[i];
        j = i-1;
        mientras(j≥P && T[j]>A);
            T[j+1] = T[j];
            j--;
        T[j+1] = A;
```

Observaciones:

- El trabajo del bucle interno depende de la entrada.
- El trabajo sobre una entrada σ será:

$$n_{is}(\sigma) = \sum_{i=2}^{N} n_{is}(\sigma, i)$$

- Además: $1 \leq n_{is}(\sigma, i) \leq i-1$

$$n_{is}(\sigma) = \sum_{i=2}^{N} ? = \sum_{i=2}^{N} \underbrace{n_{is}(\sigma, i)}_{1 \leq n_{is}(\sigma,i) \leq i-1} \Rightarrow$$

$$\Rightarrow \sum_{i=2}^{N} 1 \leq \sum_{i=2}^{N} n_{is}(\sigma, i) \leq \sum_{i=2}^{N} i-1 \equiv \sum_{1}^{N-1} j \Rightarrow$$

$$\Rightarrow N-1 \leq \sum_{i=2}^{N} n_{is}(\sigma, i) \leq \frac{N(N-1)}{2} = \frac{N^2-N}{2}$$

- CASO PEOR
  - Paso 1: por lo anterior $\forall \sigma \in \Sigma_N$, $n_{is}(\sigma) \leq \frac{N(N-1)}{2}$
  - Paso 2: $n_{is}([N, N-1, N-2, \dots, 2, 1]) = \frac{N(N-1)}{2}$

  $\Rightarrow \boxed{W_{is}(N) = \frac{N(N-1)}{2}}$

- CASO MEJOR
  - Paso 1: por lo anterior $\forall \sigma \in \Sigma_N$, $n_{is}(\sigma) \geq N-1$
  - Paso 2: $n_{is}([1, 2, \dots, N-1, N]) = N-1$

  $\Rightarrow \boxed{B_{is}(N) = N-1}$

- CASO MEDIO

Empezamos con la definición:

$$A_{is}(N) = \sum_{\sigma \in \Sigma_N} p(\sigma) \cdot n_{is}(\sigma) = \frac{1}{N!} \sum_{\sigma \in \Sigma_N} n_{is}(\sigma) = \frac{1}{N!} \sum_{\sigma \in \Sigma_N} \sum_{i=2}^{N} n_{is}(\sigma, i) =$$

$$= \sum_{i=2}^{N} \frac{1}{N!} \sum_{\sigma \in \Sigma_N} n_{is}(\sigma, i) = \sum_{i=2}^{N} \boxed{A_{is}(N, i)} \rightarrow$$ nº medio de operaciones que realiza IS en la iteración i.

$$A_{is}(N, i) = \sum \#cdc \cdot prob(hacerlas) \simeq \frac{1}{i}(1 + 2 + \cdots + i-1)$$

$$\left\lfloor \frac{N}{2} \right\rfloor \leq n_{combinar}(T, \underset{1}{P}, M, \underset{n}{U}) \leq N-1$$

$$\log T_{izq.} \quad M = \left\lfloor \frac{1+N}{2} \right\rfloor$$

$$\log T_{der} = \left\lfloor \frac{N}{2} \right\rfloor$$

caso peor combinar

$$n_{MS}(\sigma) \leq \underbrace{n_{MS}(\sigma_{izq.})}_{\wedge| \; W_{MS}\left(\left\lceil \frac{N}{2} \right\rceil\right)} + \underbrace{n_{MS}(\sigma_{der.})}_{\wedge| \; W_{MS}\left(\left\lfloor \frac{N}{2} \right\rfloor\right)} + (N-1) \Rightarrow$$

$$W_{MS}\left(\left\lceil \frac{N}{2} \right\rceil\right) + W_{MS}\left(\left\lfloor \frac{N}{2} \right\rfloor\right) + (N-1)$$

$$\Rightarrow n_{MS}(\sigma) \leq n_{MS}(\sigma_{izq.}) + n_{MS}(\sigma_{der}) + (N-1) \leq W_{MS}\left(\left\lceil \frac{N}{2} \right\rceil\right) + W_{MS}\left(\left\lfloor \frac{N}{2} \right\rfloor\right) + (N-$$

$$\Rightarrow W_{MS}(N) \leq W_{MS}\left(\underset{N}{\left\lceil \frac{N}{2} \right\rceil}\right) + W_{MS}\left(\underset{N}{\left\lfloor \frac{N}{2} \right\rfloor}\right) + N-1$$

caso base: $W_{MS}(1) = 0 \leftarrow$ trabajo nulo

tabla 1 elemento

Suponemos que $N = 2^k$.

$$\begin{cases} W_{MS}(N) \leq W_{MS}\left(\frac{N}{2}\right) + W_{MS}\left(\frac{N}{2}\right) + N-1 = 2W_{MS}\left(\frac{N}{2}\right) + N-1 \\ W_{MS}(1) = 0 \end{cases}$$

$$W_{MS}(N) \leq N-1 + 2\left(\frac{N}{2} - 1 + 2W_{MS}\left(\frac{N}{N^2}\right)\right) = N + N - 1 - 2 + 2^2 W_{MS}\left(\frac{N}{2^2}\right) \leq$$

$$\leq N + N - 1 - 2 + 2^2\left(\frac{N}{2^2} - 1 + 2W_{MS}\left(\frac{N}{2^3}\right)\right) = N + N + N - 1 - 2 - 2^2 + 2^3 \cdot W_{MS}\left(\frac{N}{2^3}\right)$$

$\Rightarrow W_{MS}(N)$ cuando $N = 2^k$

$$W_{MS}(N) \leq K.N - \sum_{j=0}^{K-1} 2^j + 2^K \cancel{W\left(\frac{N}{2^K}\right)}^{W_{MS}(1)=0} = K.N - \sum_{j=0}^{K-1} 2^j =$$

$$\boxed{\frac{U.R - P}{R-1}}$$

$$= KN - \frac{2^K - 1}{1} = \underbrace{K}_{\log_2 N} N - \underbrace{2^K}_{N} + 1 = N \log N - N + 1$$

$$\Rightarrow \boxed{W_{MS}(N) \leq N.\log N + O(N)}$$

$$n_{MS}(\sigma) \geq \underbrace{n_{MS}(\sigma_{izq.})}_{\geq\, B_{MS}\left(\lceil \frac{N}{2} \rceil\right)} + \underbrace{n_{MS}(\sigma_{der})}_{\geq\, B_{MS}\left(\lfloor \frac{N}{2} \rfloor\right)} + \left\lfloor \frac{N}{2} \right\rfloor$$

$$B_{MS}\left(\left\lceil \frac{N}{2} \right\rceil\right) + B_{MS}\left(\left\lfloor \frac{N}{2} \right\rfloor\right) + \left\lfloor \frac{N}{2} \right\rfloor$$

$$\Rightarrow \begin{cases} B_{MS}(N) \geq B_{MS}\left(\left\lceil \frac{N}{2} \right\rceil\right) + B_{MS}\left(\left\lceil \frac{N}{2} \right\rceil\right) + \left\lfloor \frac{N}{2} \right\rfloor \\ B_{MS}(1) = 0 \longrightarrow \text{caso base} \end{cases}$$

Suponemos $N = 2^k$.

$$B_{MS}(N) \geq \frac{N}{2} + 2B_{MS}\left(\frac{N}{2}\right) \geq \frac{N}{2} + 2\left(\frac{N/2}{2} + 2B_{MS}\left(\frac{N/2}{2}\right)\right) =$$

$$= \frac{N}{2} + \frac{N}{2} + 2^2 B_{MS}\left(\frac{N}{2^2}\right) = \frac{2N}{2} + 2^2 . B_{MS}\left(\frac{N}{2^2}\right) \leq$$

$$\leq K . \frac{N}{2} + 2^K \cancel{B_{MS}\left(\frac{N}{2^K}\right)}^{B_{MS}(1)=0} = \overset{\log_2 N}{K} . \frac{N}{2} = \frac{N \log N}{2}$$

$$\Rightarrow \boxed{B_{MS}(N) \geq \frac{1}{2} N \log N}$$

Trabajo partir:

$$n_{partir}(N) = \sum_{i=2}^{N} 1 = N-1$$

CASO PEOR QUICKSORT

$$n_{QS}(\sigma) = N-1 + \underset{i-1}{n_{QS}(\sigma_{izq.})} + \underset{N-i}{n_{QS}(\sigma_{der.})} \leq$$

$\boxed{1 \leq i \leq N}$

$\boxed{\sigma(1) = i}$

$$\leq N-1 + W_{QS}(i-1) + W_{QS}(N-i) \leq$$

$$\leq N-1 + \max\left\{W_{QS}(i-1) + W_{QS}(N-i)\right\} \quad \text{para } i \in [1,N].$$

$$\Rightarrow W_{QS}(N) \leq N-1 + \max\left\{W_{QS}(i-1) + W_{QS}(N-i)\right\} \quad \text{para } i \in [1,N]$$

$$W_{QS}(N) = \frac{N(N-1)}{2} \qquad (\text{ej. } \sigma = [1 \text{ — } N])$$

CASO MEDIO QUICKSORT

$$\underbrace{\sigma_{izq}}_{i-1} \quad i \quad \underbrace{\sigma_{der}}_{N-i}$$

$$n_{QS}(\sigma) = N-1 + n_{QS}(\sigma_{izq}) + n_{QS}(\sigma_{der}) \simeq$$

$$\simeq N-1 + \frac{1}{N}\sum_{i=1}^{N}\left[A(i-1) + A(N-i)\right]$$

Sabiendo $A_{QS}(1) = 0$

$$\Rightarrow A_{QS}(N) = N-1 + \frac{1}{N}\sum_{i=1}^{N}\left[A(i-1) + A(N-i)\right]$$

$\cancel{A(0) + A(1) + \cdots + A(N-2) + A(}$

$\Rightarrow A(0) + A(1) + \cdots + A(N-2) + A(N-$

$A(N-1) + A(N-2) + \cdots + A(1) + A($

$$\Rightarrow A_{QS}(N) = N-1 + \frac{2}{N}\sum_{j=1}^{N-1} A(j) \Rightarrow$$

multipl. por N

$$\Rightarrow N \cdot A_{QS}(N) = N^2 - N + 2\sum_{j=1}^{N-1} A(j) \; (1) \Rightarrow$$

astucia

$$\Rightarrow (N-1) \cdot A_{QS}(N-1) = (N-1)(N-2) + 2\sum_{1}^{N-2} A(i) \quad (2)$$

Restamos $(1) - (2)$

$$\Rightarrow N \cdot A(N) - (N-1)\cdot A(N-1) = 2(N-1) + 2 \cdot A(N-1) \Rightarrow$$

$$\Rightarrow \boxed{N \cdot \underbrace{A(N)} = 2(N-1) + (N+1)\cdot \underbrace{A(N-1)}} \Rightarrow \text{dividimos por } N(N+1)$$

recurrencia !

$$\frac{A(N)}{N+1} = \frac{2(N-1)}{N(N+1)} + \frac{A(N-1)}{N}$$

$$\overset{\shortparallel}{B(N)}$$

$$B(N) = \frac{2\cdot(N-1)}{N(N+1)} + B(N-1) \qquad ; \quad B(1) = 0$$

Deshacemos la recurrencia:

$$B(N) = \frac{2\cdot(N-1)}{N\cdot(N+1)} + \frac{2(N-2)}{(N-1)N} + \frac{2(N-3)}{(N-2)(N-1)} + B(N-3) =$$

$$= \frac{2(N-1)}{N(N+1)} + \frac{2(N-2)}{(N-1)\cdot N} + \cdots + \frac{2j}{(j+1)(j+2)} + B(j) =$$

$$= \frac{2(N-1)}{N(N+1)} + \frac{2(N-2)}{(N-1)\cdot N} + \cdots + \frac{2}{2\cdot 3} + \underbrace{\cancel{B(1)}}_{0} = 2\sum_{j=1}^{N-1} \frac{j}{(j+1)(j+2)} =$$

$$\simeq 2\sum_{j=1}^{N-1} \frac{1}{j} \sim 2 \log N + O(1)$$

$$B(N) = \frac{A(N)}{N+1} \sim 2\log N \Rightarrow \boxed{A(N) = 2N\log N + O(N)}$$