1.- TAEs DE ALGORITMOS BÁSICOS

1. Algoritmo tradicional de multiplicación de matrices cuadradas = MM

EJERCICIO

```
matriz MM(matriz A, matriz B, dim N)
    para i de 1 a N:
        para j de 1 a N:
            c[i,j] = 0;
            para K de 1 a N:
                c[i,j] += a[i,K] * b[K,j];

    devolver C
```

OB = la multiplicación del bucle más interno.

$$n_{MMul}(A,B) = \sum_{1}^{N} \sum_{1}^{N} \sum_{1}^{N} 1 = N^3$$

2. Algoritmo que encuentra el mínimo de una tabla:

EJERCICIO

```
indice Min(tabla T, indice P, indice U)
    Min = T[P];
    iMin = P;
    para i de P+1 a U:
        si T[i] < Min:
            Min = T[i];
            iMin = i;
    devolver iMin;
```

OB = la comparación de claves.

$$n_{Min}(T,P,U) = \sum_{P+1}^{U} 1 = U-P$$

$$n_{Min}(T,1,N) = \sum_{2}^{N} 1 = N-2+1 = N-1$$

$\boxed{4.}$ a)

IERCICIO

```
sum = 0
for(i=1 ; i≤n ; i++)
    sum++;
```

OB: la comparación $i \leq n$ o la suma.

$$n_a = \sum_{i=1}^{N} 1 = N$$

b)

```
sum = 0
for(i = 1 ; i<=n ; i++)
    for(j=0; j<n ; j++)
        sum++;
```

OB: la comparación del bucle interno o el incremento.

$$n_b = \sum_{i=1}^{n} \sum_{j=0}^{n-1} 1 = \sum_{i=1}^{n} n = N^2$$

c)

```
sum = 0
for(i=1; i≤n ; i++)
    for(j=0 ; j< n^2 ; j++)
        sum++;
```

OB: la comparación del bucle interno o el incremento.

$$n_c = \sum_{i=1}^{n} \sum_{j=0}^{n^2-1} = \sum_{i=1}^{n} n^2 = N^3$$

$\boxed{5.}$

ROBLEMA

a)

```
sum = 0
for(i= 1 ; i≤n ; i++)
    for(j=0 ; j<i ; j++)
        sum++;
```

OB: la comparación del bucle interno o el incremento.

$$n_a = \sum_{i=1}^{N} \sum_{j=0}^{i-1} 1 = \sum_{i=1}^{N} i = \frac{N(N+1)}{2} = \frac{N^2+N}{2} = \frac{N^2}{N} + O(N$$

b)

```
sum = 0;
for(i=1 ; i≤n ; i++)
    for(j=0; j<i^2 ; j++)
        for(k=0; k<j; k++)
            sum++;
```

OB: la comparación del bucle interno o el incremento.

$$n_b = \sum_{i=1}^{N} \sum_{j=0}^{i^2-1} \sum_{k=0}^{j-1} 1 = \sum_{i=1}^{N} \sum_{j=0}^{i^2-1} j = \sum_{i=1}^{N} \frac{(i^2-1)i^2}{2} =$$

$$= \sum_{i=1}^{N} \frac{i^4 - i^2}{2} \simeq \sum_{i=1}^{N} i^4 + O(i^2) \text{ MAL}$$

c)

```
sum = 0
for(i= 1; i≤n; i++)
    for(j= 0 ; j<i^2; j++)
        if(j%i==0)
            for(k=0; k<j; k++)
                sum++;
```

solo se hace 1 vez pq en el primer caso

OB: la comparación del if.

$$n_c = 1 + \sum_{i=1}^{n} \sum_{j=0}^{i^2-1} 1 = 1 + \sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} = N^3 \cancel{+ O(N^2}$$

j%i==0 solo se verifica MAL

3. EJERCICIO

$$n_{pot}(x,n) = N-1$$

$$n_{pot_{++}}(x,n) = O(\lg N)$$

5. PROBLEMA

a) $$\sum_{1}^{N}\sum_{j=0}^{i-1} 1 = \sum_{1}^{N} i = \frac{N(N+1)}{2} = \frac{N^2+N}{2} = \boxed{\frac{N^2}{2} + O(N)}$$

b) $$\sum_{i=1}^{n}\sum_{j=0}^{i^2-1}\sum_{k=0}^{j-1} 1 = \sum_{i=1}^{n}\sum_{0}^{i^2-1} j = \sum_{i=1}^{n}\frac{(i^2-1)i^2}{2} = \frac{1}{2}\sum_{1}^{n} i^4 - \frac{1}{2}\sum_{1}^{n} i^2 =$$

$$= \frac{1}{2}\left(\frac{n^5}{5} + O(n^4)\right) - \frac{1}{2}\left(\frac{n^3}{3} + O(n^2)\right) = \frac{n^5}{10} + O(n^4) - \left(\frac{n^3}{6} + O(n^2)\right)$$

$$= \boxed{\frac{n^5}{10} + O(n^4)}$$

c) $$\sum_{1}^{n}\sum_{j=0}^{i^2-1}\sum_{k=0}^{j-1} = \sum_{1}^{n}\left[0 \times i + 1 \times i + 2 \times i + \dots + (i-1) \times i\right] =$$

$$= \sum_{1}^{n} i(1+2+\dots+i-1) = \sum_{1}^{n} i\,\frac{(i-1)\cdot i}{2} = \frac{1}{2}\sum_{i=1}^{n} i^3 - \frac{1}{2}\sum_{i=1}^{n} i^2 =$$

$$= \frac{n^4}{8} + O(n^3) - \left(\frac{n^3}{6} + O(n^2)\right) = \boxed{\frac{n^4}{8} + O(n^3)}$$

2.- CRECIMIENTO DE FUNCIONES

7. Ejercicio

a)

	N=10	N=100	N=1000	N=10000	N=100000	N=1000000
$10000N$	100 000	1000000	10 000 000	100 000 000	1 000 000 000	10 000 000 000
$1000N.\log_{10}N$	10 000	200 000	3 000 000	40 000 000	500 000 000	6 000 000 000
$100N^2$	10 000	1000 000	100 000 000	10^{10}	10^{12}	10^{14}
$10N^3$	10 000	10 000 000	10^{10}	10^{13}	10^{16}	10^{19}
$10^{-3}.10^{N/10}$	10^{-2}	10^{7}	10^{97}	10^{997}	10^{9997}	10^{99997}

b) $10000N.10^{-6}=60 \Rightarrow N=\frac{60.10^6}{10000}=6000$ algoritmo 1

$1000N.\log_{10}N.10^{-6}=60 \Rightarrow N\log_{10}N=\frac{60.10^6}{1000} \Rightarrow N\log_{10}N=60000 \Rightarrow$

wolfram alpha

$\Rightarrow N\simeq 14426$ algoritmo 2

$100N^2.10^{-6}=60 \Rightarrow N=+\sqrt{\frac{60.10^6}{100}}=774'597 \simeq 774$ alg 3

$10N^3.10^{-6}=60 \Rightarrow N=+\sqrt[3]{\frac{60.10^6}{10}}=181'712 \simeq 181$ alg 4

$10^{-3}.10^{N/10}.10^{-6}=60 \Rightarrow 10^{N/10}=60.10^9 \Rightarrow \frac{N}{10}=\log_{10}(60.10^9)$

$\Rightarrow N=10.\log_{10}(60.10^9)=107'782 \simeq 107$ alg 5

c) algoritmo 1: $10000N.10^{-6}=86400 \Rightarrow N=\frac{86400.10^6}{10000}=8'64.10^6$

algoritmo 2: $1000N.\log_{10}N.10^{-6}=86400 \Rightarrow N.\log N=\frac{86400.10^6}{1000} \Rightarrow$

wolfram alpha

$\Rightarrow N.\log N=86400.10^3 \Rightarrow N\simeq 1'22.10^7$

algoritmo 3: $100N^2.10^{-6}=86400 \Rightarrow N=+\sqrt{86400.10^4}\simeq 29393'88\simeq 29393$

algoritmo 4: $10N^3.10^{-6}=86400 \Rightarrow N=+\sqrt[3]{\frac{86400.10^6}{10}}\simeq 2051'97\simeq 2051$

algoritmo 5: $10^{-3}.10^{N/10}.10^{-6}=86400 \Rightarrow \frac{N}{10}=\log_{10}(86400.10^9) \Rightarrow$

$\Rightarrow N=10\log_{10}(8'64.10^{13})\simeq 1533'02 \simeq 1533$

8.

EJERCICIO	$N=100$	$N=1000$	$N=10000$	$N=1000000$
N	100	1000	10000	$1000000 = 10^6$
$\sqrt{N}$	10	$10\sqrt{10} \simeq 31'62$	100	1000
$N^{1'5}$	1000	31622'78	$1000000 = 10^6$	10^9
N^2	$10000 = 10^4$	$1000000 = 10^6$	$100000000 = 10^8$	10^{12}
$N\log N$	460'52	6907'76	92103'4	$13815510'56 = 1'38.10^7$
$N\log\log N$	152'72	1932'64	22203'27	$2625791'9 = 2'6.10^7$
$N\log^2 n$	2120'76	47717'08	848303'7	$190868332 = 1'9.10^8$
$2/n$	$1/50 = 0'02$	$1/500 = 0'002$	$1/5000 = 0'0002$	$1/500000 = 0'000002$
$2^{n/2}$	$1'13.10^{15}$	$3'27.10^{150}$	$1'41.10^{1505}$	$9'95.10^{150514}$
37	37	37	37	37
$N^2\log N$	46051'7	6907755'28	$9'2.10^8$	$1'61.10^{13}$
N^3	$1000000 = 10^6$	10^9	10^{12}	10^{15}

Orden descendente: $2^{n/2}$, N^3, $N^2\log N$, N^2, $N^{1'5}$, $N\log^2 N$, $N\log\log N$, $N\log N$, N, $\sqrt{N}$, 37, $2/N$

intercambiar

10.

EJERCICIO $W_A(N) = O(1000\,N\log_{10}(N)) \Rightarrow W_A(N) \leq 1000N\log_{10}(N)$

$W_B(N) = O(N^2) \Rightarrow W_B(N) \leq N^2$

Para $N=10 \begin{cases} 1000N\log_{10}(N) = 10000 = 10^4 \\ N^2 = 100 = 10^2 \end{cases}$; Para $N=10000 \begin{cases} 1000N\log_{10}(N) = 40\,000\,000 = 4.10^7 \\ N^2 = 100\,000\,000 = 10^8 \end{cases}$

$\Rightarrow$ Para n^{os} pequeños es mejor $N^2 \Rightarrow$ es mejor $W_B(N)$

$\Rightarrow$ Para n^{os} grandes es mejor $1000N.\log_{10}(N) \Rightarrow$ es mejor $W_A(N)$

¿cuál es la "frontera" entre lo "pequeño" y lo "grande"?

wolfram alpha

$N^2 = 1000N.\log_{10}(N) \Rightarrow N = 3550'26$

n^{os} pequeños $\leq 3550 <$ n^{os} grandes

14. EJERCICIO $T_1 \sim f$ y $T_2 \sim f$

a) $T_1 + T_2 \sim f$

$T_1 \sim f$ quiere decir que $\frac{T_1}{f} \to 1$

$T_2 \sim f$ " " " $\frac{T_2}{f} \to 1$

Si $T_1 + T_2 \sim f \Rightarrow \frac{T_1+T_2}{f} \to 1$ pero $\left.\frac{T_1+T_2}{f} = \underbrace{\frac{T_1}{f}}_{\downarrow 1} + \underbrace{\frac{T_2}{f}}_{\downarrow 1}\right) \to 2$ ※

b) $T_1 - T_2 = o(f)$

$\left.\frac{T_1 - T_2}{f} = \underbrace{\frac{T_1}{f}}_{\downarrow 1} - \underbrace{\frac{T_2}{f}}_{\downarrow 1}\right) \to 0$ ✓

c) $T_1 . T_2 \sim f^2 \Rightarrow \frac{T_1 T_2}{f f} = \frac{T_1}{f} \cdot \frac{T_2}{f} \to 1$ ✓

15. EJERCICIO

a) $\left.\frac{T_1+T_2}{2f} = \underbrace{\frac{T_1}{2f}}_{\downarrow \frac{1}{2}} + \underbrace{\frac{T_2}{2f}}_{\downarrow \frac{1}{2}}\right) \to 1 \Rightarrow T_1 + T_2 \sim 2f$ ✓

b) $T_1^2 - T_2^2 = o(f)$

$\frac{T_1^2 - T_2^2}{f} = T_1 . \underbrace{\frac{T_1}{f}}_{\downarrow 1} - T_2 . \underbrace{\frac{T_2}{f}}_{\downarrow 1}$

Vemos que no tiene futuro y buscamos un contraejemplo

$\left.\begin{array}{l} T_1 = N+1 \\ T_2 = N \\ f = N \end{array}\right\}$ $T_1^2 - T_2^2 = (N+1)^2 - N^2 = 2N+1$

$\Rightarrow \frac{T_1^2 - T_2^2}{f} = \frac{2N+1}{N} \to 2 \neq 1$

$T_1^2 - T_2^2 \not\sim f$

c) $\frac{T_1}{T_2} \sim 1$?

$$\frac{T_1/T_2}{1} = \underbrace{\frac{T_1}{f}}_{\downarrow\, 1} \cdot \frac{1}{\underbrace{T_2/f}_{\downarrow\, 1}} \Big) \to 1 \checkmark$$

$\boxed{16.}$ $T_1 = O(f)$ y $T_2 = O(f)$

ROBLEMA

1) ¿$T_1 + T_2 = O(f)$?

Como $T_1 + T_2 = O(2f) = O(f)$ ✓

2) ¿$T_1 - T_2 = o(f)$?

contraejemplo: $T_1 = 2N$ $\quad T_1 - T_2 = N \neq o(N)$

$T_2 = N$

$f = N$

3) ¿$\frac{T_1}{T_2} = O(1)$?

contraejemplo: $\left.\begin{array}{l} T_1 = N^2 \\ T_2 = N \\ f = N^2 \end{array}\right\} \quad \frac{T_1}{T_2} = N \neq O(1)$

4) ¿$T_1 = O(T_2)$?

contraejemplo: $\left.\begin{array}{l} T_1 = N^2 \\ T_2 = N \\ f = N^2 \end{array}\right\} \quad T_1 \neq O(T_2)$

13. EJERCICIO AVANZADO

$f = N\log N + 100N$

¿crecimiento teórico?
¿crecimiento real?

$N\log N > 100N$

$\log_{10} N > 100$

$N > 10^{100}$

crec. teórico $= f(N) = O(N\log N)$

creci. real $\rightarrow$ nunca se va alcanzar

10. EJERCICIO

$\downarrow$

hecho anteriormente

$W_A = O(1000\,N \log_{10} N)$

$W_B = O(N^2)$

N	W_A	W_B	¿MEJOR?
100	2×10^5	10^4	B
10^3	3×10^6	10^6	B
10^4	4×10^7	10^8	A

8.

$\frac{2}{N} < 37 < \sqrt{N} < N < N\log\log N < N\log N < N\log^2 N < N^{1'5} <$

$< N^2 < N^2\log N < N^3 < 2^{n/3}$

12. ¿$\lceil N \rceil = \Theta(N)$?

$1 \leq \frac{\lceil N \rceil}{N} \leq \frac{N+1}{N} \Rightarrow \frac{\lceil N \rceil}{N} \longrightarrow 1$

$\downarrow \qquad \downarrow$

$1 \qquad 1$

$\lceil N \rceil \sim N \rightarrow \lceil N \rceil = N + O(1)$

Igual para: $\lfloor N \rfloor \sim N$
$\rightarrow \lfloor N \rfloor = N + O(1)$

$\boxed{20.}$ Para cualquier f y cualquier constante C :

- ¿ $f(Cn) = O(f(n))$?

contraejemplo: $f(n) = e^n$

$$f(2n) = e^{2n} = (e^n)^2 \neq O(e^n)$$

- ¿ $f(n+C) = O(f(n))$?

$f(n) = e^n$

$f(n+2) = e^{n+2} = e^2 e^n \overset{\checkmark}{=} O(e^n)$ parece que cumple lo que antes no.

$$f(n) = n! \sim \left(\frac{n}{e}\right)^n$$

$$f(n+1) = (n+1)! = (n+1)n! = (n+1)f(n) \neq O(f(n))$$

3.- CASOS PEOR, MEJOR Y MEDIO DE ALGORITMOS BÁSICOS

EJERCICIO 24. tabla T, se sabe que: $p(k == T[i]) = \frac{1}{C_N} \cdot \frac{\log i}{i}$

donde $C_N = \sum_1^N \frac{\log i}{i}$ Calcular $A_{BL}(N)$

$$A_{BL}(N) = \sum_{i=1}^N n_{BL}(k = T[i])\, p(k == T[i]) = \sum_{i=1}^N i \frac{1}{C_N} \cdot \frac{\log i}{i} =$$

$$= \frac{1}{C_N} \sum_{i=1}^N i \cdot \frac{\log i}{i} = \frac{1}{C_N} \sum_{i=1}^N \log i = \frac{S_N}{C_N}$$

$$C_N = \sum_1^N \frac{\log i}{i}$$

esto visto en teoría

$$S_N = \sum_1^N \log i \sim N \cdot \log N$$

porque $f(x)$ es decreciente

$$\int_0^N \frac{\log x}{x} dx \geq C_N \geq \int_1^{N+1} \frac{\log x}{x} dx \Rightarrow \frac{1}{2}\left[\log^2 x\right]_0^N \geq C_N \geq \cdots$$

$\log 0 = \infty$

Entonces escribimos: $C_N = \frac{\log 1}{1} + \sum_2^N \frac{\log i}{i} = \sum_2^N \frac{\log i}{i}$

$$\int_1^N \frac{\log x}{x} dx \geq C_N \geq \int_2^{N+1} \frac{\log x}{x} dx \Rightarrow \frac{1}{2}\left[\log^2 x\right]_1^N \geq C_N \geq \frac{1}{2}\left[\log^2 x\right]_2^{N+1} \Rightarrow$$

$$\Rightarrow \frac{1}{2}\log^2 N \geq C_N \geq \frac{1}{2} \cdot \log^2(N+1) - \frac{1}{2}\log^2 2$$ $\Rightarrow$ dividimos por $\frac{1}{2}\log^2 N$

$$\Rightarrow 1 \geq \frac{C_N}{\frac{1}{2}\log^2 N} \geq \underbrace{\left(\frac{\log(N+1)}{\log N}\right)^2}_{\xrightarrow[n\to\infty]{} 1} - \left(\frac{\log 2}{\log^2 N}\right)^2 \xrightarrow[n\to\infty]{} 0$$

$$\Rightarrow \boxed{C_N = \sum_1^N \frac{\log i}{i} \sim \frac{1}{2}\log^2 N}$$

$$\Rightarrow A_{BL}(N) = \frac{S_N}{C_N} \sim \frac{N \log N}{\frac{1}{2}\log^2 N} = \frac{2N}{\log N}$$

Demostración de que este cociente tiende a 1:

$$\frac{A_{BL}(N)}{\frac{2N}{\log N}} = \frac{S_N}{C_N} \cdot \frac{1}{\frac{N \cdot \log N}{\frac{1}{2}\log^2 N}} = \underbrace{\frac{S_N}{N \log N}}_{\to 1} \cdot \underbrace{\frac{1}{\frac{C_N}{\frac{1}{2}\log^2 N}}}_{\to 1}$$

(*) ya hemos visto que estas cosas tienden a 1 cuando $n \to \infty$.

21.
JERCICIO

$$S_N = \sum_{1}^{N} \frac{1}{i^{1/3}} = \sum_{i=1}^{N} f(i)$$

$$\int_0^N \frac{1}{x^{1/3}}\,dx \geq S_N \geq \int_1^{N+1} \frac{1}{x^{1/3}}\,dx \Rightarrow \frac{3}{2}\left[x^{2/3}\right]_0^N \geq S_N \geq \frac{3}{2}\left[x^{2/3}\right]_1^{N+1} =$$

decreciente

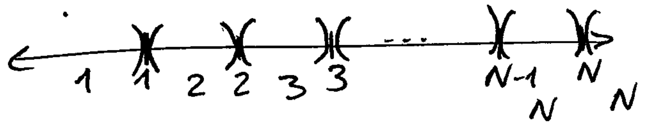
$$\Rightarrow \frac{3}{2}N^{2/3} \geq S_N \geq \frac{3}{2}(N+1)^{2/3} - \frac{3}{2}$$

¿ $S_N \sim \frac{3}{2}N^{2/3}$? $\longrightarrow$ dividimos por $\frac{3}{2}N^{2/3}$ $\longrightarrow$

$$\longrightarrow 1 \geq \frac{S_N}{\frac{3}{2}N^{2/3}} \geq \underset{\downarrow \atop 1}{\frac{(N+1)^{2/3}}{N^{2/3}}} - \underset{\downarrow \atop 0}{\frac{1}{N^{3/2}}} \Rightarrow \frac{S_N}{\frac{3}{2}N^{2/3}} \to 1 \Rightarrow S_N \sim \frac{3}{2}N^{2/3}$$

23.
JERCICIO

$$E_{BLO}(N) = \underbrace{\{1, \dots, N\}}_{\text{éxito}} \cup \overbrace{\{K_{(-\infty,1)}, K_{(1,2)}, \dots, K_{(N-1,N)}, K_{(N,\infty)}\}}^{\text{fracaso}}$$



$$|E_{BL}(N)| = N + N + 1 = 2N+1$$

$$P(K) = \frac{1}{2N+1}$$

$$A_{BLO}(N) = \frac{1}{2N+1}\left\{\underbrace{\sum_{K=1}^{N}(1+K)}_{\text{trabajo\_éxito}} + \underbrace{\sum_{K=1}^{N}(K+1) + N}_{\text{trabajo\_fracaso}}\right\} =$$

$$= \frac{1}{2N+1}\left(N + \frac{N(N+1)}{2} + N + \frac{N(N+1)}{2} + N\right) =$$

$$= \frac{1}{2N+1}\left(N^2 + N + 3N\right) = \frac{1}{2N+1}\left(N^2 + 4N\right) =$$

$$= \frac{N^2+4N}{2N+1} = \frac{N}{2} + O(1)$$

4.- EVOLUCIÓN DE ALGORITMOS LOCALES

29. lista = [20, 3, 10, 6, 8, 2, 13, 17] $\xrightarrow{\text{para resumir}}$ lista = [20, 3, 10, 6, 8]

JERCICIO

BURBUJA

iteración ext. 1: 3 10 6 8 | 20

iteración ext. 2: 3 6 8 | 10 20

iteración ext. 3: 3 6 | 8 10 20

iteración ext. 4: 3 | 6 8 10 20

INSERCCIÓN lista = [20 | 3, 10, 6, 8] 1 comp

iteracción 1: 3 20 | 10 6 8 2 comp.

iteracción 2: 3 10 20 | 6 8 3 comp.

teracción 3: 3 6 10 20 | 8 3 comp.

teracción 4: 3 6 8 10 20

9 comp. totales

33. $\underbrace{(6\ 1\ 5\ 2\ 4\ 3)}_{\sigma}$ Con un algoritmo local ¿cuántas cdcs efectuará como mínimo sobre ellas?

Como es un algoritmo local $\Rightarrow n_A(\sigma) \geq inv(\sigma) = 5+3+1 = 9$ cdc

inacabado

35. $\sigma = (\overbrace{2K+1, 2K+2, \ldots, 3K}^{\text{Bloque 1}} \mid \overbrace{2K, 2K-1, \ldots, K+2, K+1}^{\text{Bloque 2}} \mid \overbrace{1, 2, \ldots, K}^{\text{Bloque 3}})$

2K+1 está invertido con (2K, K)
y lo mismo le pasa
a 2K+2, 2K+3, …, 3K

$\boxed{\text{Bloque 3} = 0}$

$n_A(\sigma) \geq inv(\sigma) = inv(\text{bloque1}) + inv(\text{bloque2}) + inv(\text{bloque 3}) =$

Bloque 1 = K . 2K inversiones

Bloque 2 = $\sum_{i=K}^{2K-1} j \equiv$ inversiones internas bloque 2 = $\frac{K(K-1)}{2}$ (tabla completamente desordenada)

inversiones externas bloque 2 : $K \cdot K = K^2$

34. $N=2K$

$\circledast$ $n_A(\sigma) \geq inv(\sigma)$

$$\sigma = (\overbrace{2K, 2K-1, 2K-2, \dots, K+1,}^{K} | \overbrace{1, 2, \dots, K}^{K})$$

$$\left.\begin{array}{l} Int_1 = \frac{K(K-1)}{2} \\ Ext_1 = K \cdot K = K^2 \\ Int_2 = 0 \\ Ext_2 = 0 \end{array}\right\} \quad \frac{K(K-1)}{2} + K^2 = \frac{K^2-K}{2} + K^2$$

Como $K = \frac{N}{2}$

$$\frac{\frac{N}{2}\left(\frac{N}{2}-1\right)}{2} + \frac{N^2}{4} = \frac{N^2}{8} - \frac{N}{4} + \frac{N^2}{4} = \frac{3N^2}{8} - \frac{2N}{8} = \boxed{\frac{3N^2-2N}{8}}$$

36. $N=4K$ $\quad n_A(\sigma) \geq inv(\sigma)$

$$\sigma = (\overbrace{3K+1, 3K+2, \dots, 4K,}^{K} | \overbrace{3K, 3K-1, \dots, 2K+2, 2K+1, 2K, 2K-1, \dots, K+2, K+1,}^{2K} | \overbrace{1, 2, \dots, K}^{K})$$

$$\left.\begin{array}{l} Ext_1 = 3K^2 \\ Int_1 = 0 \\ Ext_2 = 2K^2 \\ Int_2 = \frac{2K(2K-1)}{2} \\ Ext_3 = 0 \\ Int_3 = 0 \end{array}\right\} \quad 5K^2 + \frac{4K^2-2K}{2} = \frac{14K^2-2K}{2} = 7K^2 - K$$

Como $K = \frac{N}{4}$

$$\boxed{\frac{7N^2}{16} - \frac{N}{4}}$$

$\boxed{32.}$

1) BubleSort

2) InsertSort

3) SelectSort de mínimos

$\boxed{45.}$

$$A_{BS\_flag}(N) = \frac{1}{N!} \sum_{\sigma \in \Sigma} n_{BS\_flag}(\sigma) =$$

$$S_K = \{\sigma \in \Sigma_N : \text{BS-flag hace } K \text{ iteraciones ext. en } \sigma\}$$

$$1 \leq K \leq N-1$$

$$S_1 \rightarrow N-1$$

$$S_2 \rightarrow N-2$$

$$S_K \rightarrow N-K$$

$$= \frac{1}{N!} \sum_{K=1}^{N-1} \sum_{\sigma \in S_K} n_{BS\_flag}(\sigma) =$$

$$= \frac{1}{N!} \sum_{K=1}^{N-1} T_K \sum_{\sigma \in S_K} 1 = \sum_{K=1}^{N-1} T_K \cdot \frac{1}{N!} \sum_{\sigma \in S_K} 1 = \rightarrow \frac{\text{permut. } S_K}{N!}$$

$$T_K \rightarrow \sum_{j=K}^{N-1} j$$

$$\frac{\text{permut. } S_K}{N!} \downarrow P(S_K)$$

$=$

$\boxed{40.}$ $\sigma = (2K, 1, 2K-1, 2, 2K-2, 3, \dots, 2K-i, i+1, \dots, K+1, K)$ $N=2K$

$n_A(\sigma) \geq inv(\sigma) = 2K-1 + 0 + 2K-3 + 0 + 2K-5 + \dots + 3 + 1 =$

$= 1+3+5+ \dots + 2K-5 + 2K-3 + 2K-1$

Lo reorganizamos:

$$\frac{K}{2} \text{ veces} \left\{ \begin{array}{l} 2K-1+1 = 2K \\ 3 + 2K-3 = 2K \\ \quad\vdots \end{array} \right.$$

$\boxed{2K \cdot \frac{K}{2} = K^2} \longrightarrow$ como $N = 2K \rightarrow K = \frac{N}{2}$ $\quad \frac{N^2}{4}$

$\boxed{39.}$ $\sigma = (1, 2K, 2, 2K-1, 3, 2K-2 \dots, K, K+1)$ $\boxed{N=2K}$

$n_A(\sigma) \geq inv(\sigma) = 0 + 2K-2 + 0 + 2K-4 \dots + 2 = 2(1+2+3+\dots+K-1) =$

$= 2 \cdot \frac{K(K-1)}{2} = K^2 - K$

Como $K = \frac{N}{2} \longrightarrow inv(\sigma) = \boxed{\frac{N^2}{4} - \frac{N}{2}}$