Clase 9 - Tiempo de ejecucion

martes, 25 de abril de 2023 16:05

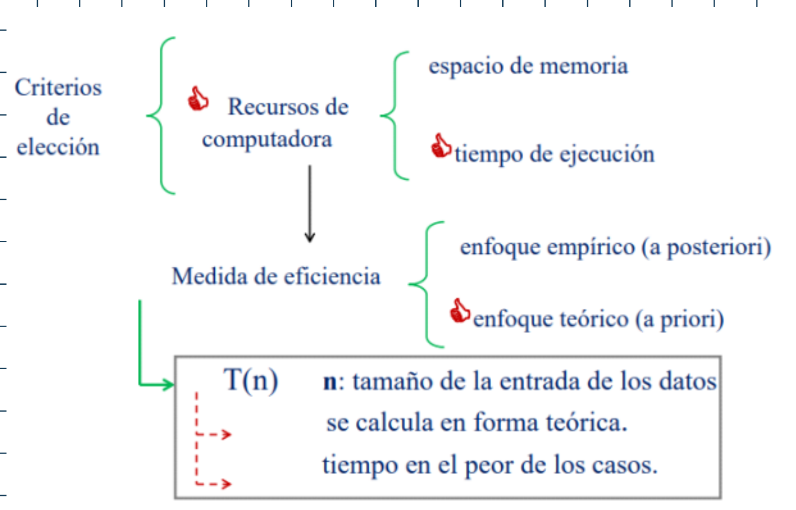
- para comparar algor. sin importar la plataforma
- mide eficacia dependiendo de la entrada.

Pasos

→ caracterizar datos → qué cosas de la entrada influyen en el T. de ejecución

→ identificar operaciones (no todas, las q' influyen)

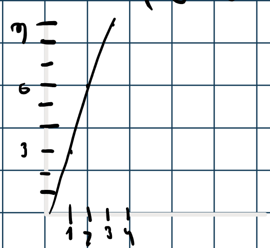
→ crear ritos



Tasa de crecimiento → q' tan rápido crece una función

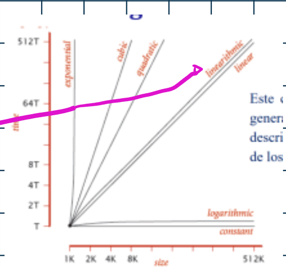
$T(n) = 3N$

→ por cada elem(n), 3 operaciones.



Notación asintótica → Descarta coeficientes constantes y los menos significativos

Ordenadas en forma creciente
1 constante — (valor siempre =)
log n
n lineal
n log n
n^2
n^3
c^n $c>1$



Considerando que un algoritmo requiere *f(n)* operaciones para resolver un problema y la computadora procesa 100 operaciones por segundo.

Si *f(n)* es:

a.- $\log_{10} n$

b.- $\sqrt{n}$

Determine el tiempo en segundos requerido por el algoritmo para resolver un problema de tamaño n=10000.

a. $\log_{10} 10000 = 4 \longrightarrow$ 100 op $\longrightarrow$ 1 seg

b $\sqrt{10000} = 100 \longrightarrow$ 1 seg

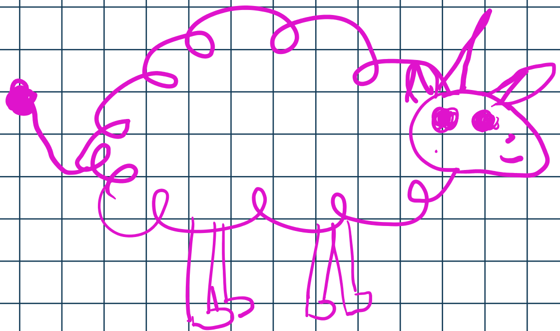
4 op $\longrightarrow$ e $\frac{4 \cdot 1}{100} = \frac{1}{25}$ seg

Suponga que Ud. tiene un algoritmo ALGO-1 con un tiempo de ejecución exacto de $10n^2$. ¿En cuánto se hace más lento ALGO-1 cuando el tamaño de la entrada n aumenta:……….?

a.- El doble

b.- El triple

??



Cálculo del Tiempo de Ejecución

➢Iteración:

a) For

Viene como parámetro

const.

```
int sum = 0;
int [] a = new int [n];
for (int i =1; i<= n ; i++ )
    sum += a[i];
```

incremento de n 1

constante 2

$$T(n) = cte_1 + \sum_{i=1}^{n} cte_2 =$$

$$= cte_1 + n * cte_2$$

$$\Rightarrow O(n)$$

Σ

a) For

Viene como parámetro

c

```
int sum = 0;
int [] a = new int [n][n];
for (int i =1; i<= n ; i++) {
    for (int j =1; j<= n ; j++)
        sum += a[i][j];
}
```

$\sum_{i=1}^{N} \sum_{j=1}^{N} O$

$T(n) = C + \sum_{i=1}^{N} N.O$

$C + N.ON$

$T(N)\ C + O.N^2 = O(n^2)$

$\sum_{I=1}^{N} 1 \neq N.I$

$\sum_{i=1}^{n} s \neq N \cdot I$

Tienen q' ser dominador uno p. poder hacerlo así.

a) For

Viene como parámetro

C
```
int [] a = new int [n];
int [] s = new int [n];
for ( int i =1; i<= n ; i++ )
    s[i] = 0;                      D
for ( int i =1; i<= n ; i++) {
   for (int j =1; j<= i ; j++)
        s[i] += a[j];
}
```

$C + \sum_{i=1}^{N} D + \sum_{i=1}^{N} \sum_{j=1}^{I} E$

$C + ND + \sum_{i=1}^{N} IE$ $\longrightarrow$ no puedo hacer NIE

$C + ND + E \cdot \frac{N(N+1)}{2}$

$C + ND + \frac{E}{2}(N^2 + N) = O(n^2)$

C
```
int [] a = new int [n];
int [] s = new int [n];
for ( int i =1; i<= n ; i++ )
    s[i] = 0;
for ( int i =1; i<= n ; i++) {
   for (int j =1; j<= i ; j++)
        s[i] += a[j];
}
```

$C + \sum_{i=1}^{n} c_2 + \sum_{i=1}^{n} \sum_{j=1}^{i} c_3$

$C + nc_2 + \sum_{1}^{n} i c_3$

$C + nc_2 + c_3 \sum_{i=1}^{n} i$

$C + nc_2 + c_2 + c_3 \frac{n(n+1)}{2}$

$\underbrace{C + nc_2 + c_2 +}_{C} \frac{c_3}{2} n^2 + \frac{c_3 n}{2}$ $\quad O(n^2)$

b) While

con tr.
```
int x= 0;
int i = 1;
while ( i <= n) {
     x = x + 1;
     i = i + 2;
}
```

$T(n) = c_1$

x = n
while x > 1
x = x/2

$1 = n$

$2 = n/2$

$3 = n/2^2$

$4 = n/2^3$

$k = n/2^{k-1}$

$k+1 =$ …

Considere el siguiente fragmento de código:

```
c. int count = 0;
   int N = a.length;
   for (int i = 0; i < N; i++)
          for (int j = 0; j < N; j++)
                 a[j]++;
```

Suponga que tarda 1 seg cuando N=3500,
¿cuánto tardará *aproximadamente* para N=35000? Justifique su respuesta.

$$\sum_{i=0}^{N} \sum_{j=0}^{N} C_2$$

$$\sum_{i=0}^{N} N\, C_2$$

$$C + N^2 C_2 \rightarrow O(n^2)$$

```
private void imparesypares(int n){
 int x=0; int y=0;

 for (int i=1;i<=n;i++)
      if (esImpar(i))
           for (int j=i;j<=n;j++)
               x++;
      else
           for (int j=1;j<=i;j++)
               y++;
}
```

```
public boolean esImpar(int unNumero){
  C  if (unNumero%2 != 0)
        return true;
     else
        return false;
}
```

$$\sum^{n}$$

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

$\rightarrow i = 1$

$j = j + 2$

$0 = 1$

$1 \rightarrow 3$

$2 \rightarrow 5$

$3 \rightarrow 7$

$4 \rightarrow 9$

$\boxed{n \rightarrow (n \cdot 2) + 1}$

$\sum^{1,2,4,?}$

```
for (i = 0; i < n; i++)
    for (j = 1; j < n; j+=n/2)
        x = x + 1;
```

(a) $O(\sqrt{n})$ (b) $O(n)$ (c) $O(n \log n)$ (d) $O(n^2)$ (e) $O(n^3)$

Considere el siguiente fragmento de código:

```
c [ int count = 0;
    int N = a.length;
    for (int i = 0; i < N; i++)
        for (int j = 0; j < N; j++)
            a[j]++;
```

Suponga que tarda 1 seg cuando N=3500,
¿cuánto tardará *aproximadamente* para N=35000? Justifique su respuesta.

$$C + \sum_{i=0}^{N-1} \left[\sum_{j=0}^{N-1} c \right]$$

$$C + \sum_{i=0}^{N-1} \left[c(N-1) \right]$$

$$C + \sum_{i=0}^{n-1} cn - c$$

$$c + n \cdot \cancel{1} \cdot (cn - \cancel{c})$$

$\cancel{c} + \cancel{c}n^2$