Análisis teórico

**1)**

**para i = 1 hasta n Hacer t1 \* n**

**j = 1 t2 \* n**

**MIENTRAS j < n Hacer t3\* n\* n**

**j = j + 3 t4 \* n \* Log**

**Fin Mientras -----**

**Fin Mientras -----**

**(t1\*n) + (t2\*n) + (t3\*n\*n)+(t4\*n\*log)**

**2)**

**PARA i = 1 HASTA n HACER t1 \* n**

**PARA j = 2\*i HASTA n HACER t2 \* n \* n**

**s = s+1 t3 \* n \* n**

**FIN PARA**

**FIN PARA**

**(t1 \* n) + (t2 \* n \* n) + (t3 \* n \*n)**

**3)**

**PARA i = 1 HASTA n HACER t1 \* n**

**j = i t2 \* n**

**MIENTRAS j < n HACER t3 \* n \* n**

**j = 2\*j t4 \* n \* Log**

**FIN MIENTRAS**

**FIN PARA**

**(t1 \* n) + (t2 \* n) + ( t3 \* n \* n) + ( t4 \* n \* log)**

**4)**

**PARA i = 1 HASTA n HACER t1 \* n**

**j = n t2 \* n**

**MIENTRAS i\*i < j HACER t3 \* n \* j**

**j = j-1 t4 \* n \* Log**

**FIN MIENTRAS**

**FIN PARA**

**(t1 \* n) + ( t2 \* n) + ( t3 \* n \* j) + (t4 \* n \* log)**

**5)**

**PARA i = 1 HASTA n HACER t1 \* n**

**j = 2 t2 \* n**

**MIENTRAS j < i HACER t3 \* n \* i**

**j = j\*j t4 \* n \* log**

**FIN MIENTRAS**

**FIN PARA**

**(t1 \* n ) + ( t2 \* n) + (t3 \* n \* i ) + ( t4 \* n \* log)**

**BIG-O(NOTACION O GRANDE)**

1. **¿Qué es la notación O grande (también conocida como Big-O)?**

R//Al tratar de calcular ***la eficiencia de un algoritmo en términos de tiempo de ejecución, independiente de cualquier programa o computadora en particular, es importante cuantificar el número de operaciones o pasos que el algoritmo requerirá***. Si se considera que cada uno de estos pasos es una unidad básica de cálculo, entonces el tiempo de ejecución de un algoritmo puede expresarse como el número de pasos necesarios para resolver el problema complicado y dependerá de cómo se implemente el algoritmo.

1. **¿Para qué se utiliza dicha notación?**

R// ***La notación nos da una manera de saber cómo se va a comportar un algoritmo*** en función de los argumentos que le pasemos y la escala de los mismo.

Por ejemplo, imagínate una función que se utiliza para localizar un elemento dentro de una lista de elementos previamente guardados. Si la documentación de la misma nos dice que una operación de tipo 0(1), quiere decir que da igual cuantos elementos haya en la lista, la operación siempre tarda lo mimo. Para ser más exactos deberíamos decir que el esfuerzo de computo necesario es el mismo.

1. **¿Qué relación tiene esta notación con la notación usada en nuestros análisis teóricos?**

R//En esencia es lo mismo porque vemos cómo se comporta el algoritmo en forma tiempo.

1. **¿Qué relación tiene esta notación con los resultados de nuestros análisis empíricos?**

R// Por lo dicho en el párrafo anterior se relacionan en que ambos son una forma de medir cómo se comporta un algoritmo.