Díaz Hernández Marcos Bryan Actividad 4 - Tarea

Analisis de complejidad.

a)
```
algoritmo (){
    for (cnt 1=0, i=1; i<=n; i++){
        for (j=1; j<=n; j++){
            cnt1++;
        }
    }
}
```

- Categoria de complejidad

Son dos ciclos for, anidados en función de n con un incremento de 1 con una instrucción por lo que cada for tiene una complejidad O(n) y al estar anidados se multiplica O(n·n) y la complejidad final es: $O(n^2)$

b)
```
algoritmo2(){
    for (cnt3=0, i=1; i<=n; i*=2){
        for (j=1; j<=n; j++)
            cnt3++; int a=0; int b=10;
        }
    }
}
```

Al probar los dos algoritmos y comprobarlos con la tabla de tiempos de ejecución que nos mostro el profesor, parecen más algoritmos logáritmicos

Son dos ciclos for donde el primero esta en función de n con un incremento con un crecimiento 2^n con $n=i$ y el segundo con un incremento lineal en función de n. Por lo que aunque el primer for tenga incremento no lineal este incremento esta en función de una operación moderada sigue siendo O(n) por lo que la complejidad es: $3*O(n^2)$

c)
```
algoritmo3(int n){
    IF (n<=0)
        return 1;
    else
        return 1 + funcion3(n-8)
```

El algoritmo hace una llamada así mismo (n-8) veces, en función de n, por lo que al llamarse un número determinado o predecible de veces es de tipo lineal y su complejidad es: O(n)

- Dudas:

¿Las declaraciones de estructuras son considerados instrucciones simples?

¿Cómo se incluye o valora el tiempo en la valoración de la complejidad de un algoritmo?

¿En caso del mejor y peor caso como se evaluan las complejidades correspondientes?