Complejidad del Taller 5

1. Insertion Sort

1.1 Código en Word

```
public static void insertionSort(int[] a){
    for (int i = 1; i < a.length; i++) {
        int comp = a[i];
        int j = i;
        while (j > 0 && a[j - 1] > comp) {
            a[j] = a[j - 1];
            j--;
        }
        a[j] = comp;
    }
}
```

1.2 Tamaño del problema (llamado también "n")

El tamaño del problema son el número de iteraciones que realizo en el ciclo.

1.3 Cuánto se demora cada línea

```
public static void insertionSort(int[] a){
    for (int i = 1; i < a.length; i++) { //Constante*n + Constante
        int comp = a[i]; //Constante*(n-1)
        int j = i; //Constante*(n-1)
        while (j > 0 && a[j - 1] > comp) { //(Constante*n + Constante)*(n-1)
            a[j] = a[j - 1]; //Constante*(n-1)
            j--; //Constante*(n-1)
        }
        a[j] = comp; //Constante*(n-1)
    }
}
```

1.4 Ecuación

$T(n) = 4C\_1*(n-1) + C\_2*n + C\_3 + (C\_4*n + C\_5)*(n-1)$

1.5 Notación O dela solución de la ecuación

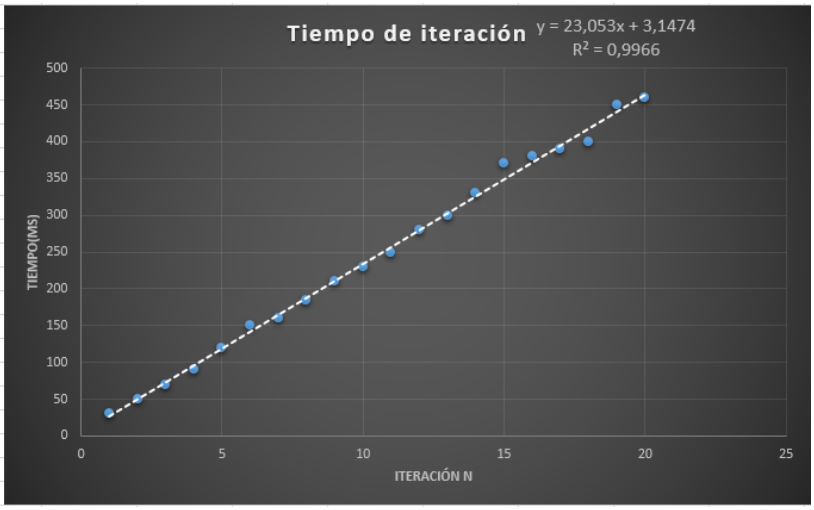
T(n) es $O(4C\_1*(n-1) + C\_2*n + C\_3 + (C\_4*n + C\_5)*(n-1))$, por definición de O
T(n) es $O(C\_2*n + (C\_4*n + C\_5)*(n-1))$, por Regla de la Suma
T(n) es O(n), por Regla del Producto

1.6 Explicado en palabras

La complejidad asintótica (es decir, para valores grandes de n) para el peor de los casos (es decir, en el que el algoritmo hace más operaciones) para el algoritmo de ordenar los elementos de un arreglo mediante ciclos es de O(n).

1.7 Grafica

Iteracion N	Tamaño arreglo	Tiempo
1	10000000	30
2	20000000	50
3	30000000	70
4	40000000	90
5	50000000	120
6	60000000	150
7	70000000	160
8	80000000	184
9	90000000	210
10	100000000	230
11	110000000	250
12	120000000	280
13	130000000	300
14	140000000	330
15	150000000	370
16	160000000	380
17	170000000	390
18	180000000	400
19	190000000	450
20	200000000	460



1.8 Análisis de los resultados

De acuerdo a los resultados obtenidos en la gráfica se puede concluir que la notación asintótica se relaciona con la gráfica, puesto que ambas tienen una ecuación lineal, la cual platea la complejidad del ejercicio. Por esto se puede observar que a la hora compilar un programa con altos índices de datos el tiempo variaría de manera constante.

2. Suma de los elementos de un arreglo

2.1 Código en Word

```
public static int suma(int[] a){
        int suma = 0;
        for(int i = 0; i < a.length; i++) {
            try {
                TimeUnit.MILLISECONDS.sleep(1/100);
            } catch (Exception e) {
            }
            suma += a[i];
        }
        return suma;
    }
```

2.2 Tamaño del problema (llamado también “n”)

El tamaño del problema son el número de iteraciones que realizo en el ciclo.

2.3 Cuánto se demora cada línea

```
public static int suma(int[] a){
    int suma = 0; // C_1
    for(int i = 0; i < a.length; i++) { // C_2 + C_3*(n+1)
        try {
            TimeUnit.MILLISECONDS.sleep(1/100);
        } catch (Exception e) {
        }
        suma += a[i];  //C_4*n
    }
    return suma; //C_5
}
```

2.4 Ecuación

$T(n) = C_1 + C_2 + C_3*(n+1) + C_4*n + C_5$

2.5 Notación O dela solución de la ecuación

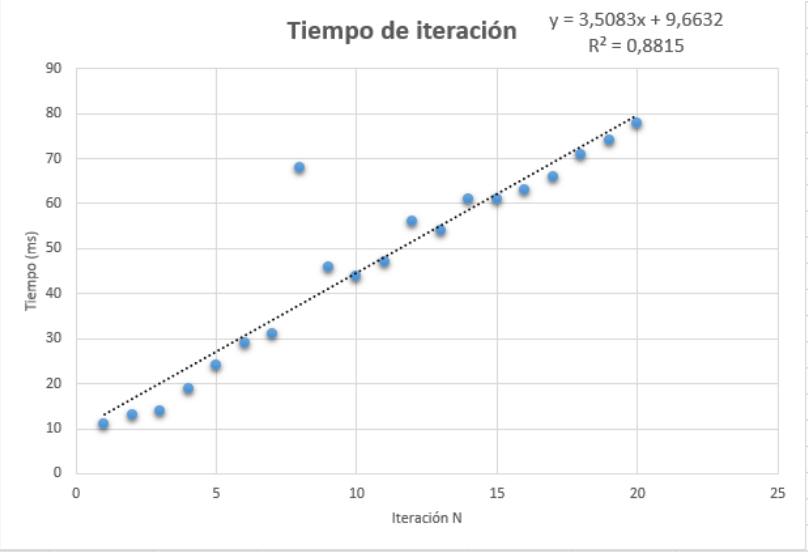
T(n) es $O(C_1 + C_2 + C_3*(n+1) + C_4*n + C_5)$, por definición de O
T(n) es $O(C_3*n)$ Regla de la suma
T(n) es O(n) Regla del producto

2.6 Explicado en palabras

La complejidad asintótica (es decir, para valores grandes de n) para el peor de los casos (es decir, en el que el algoritmo hace más operaciones) para el algoritmo de sumar los elementos de un arreglo mediante ciclos es de O(n).

2.7 Grafica

Iteracion N	Tamaño arre	Tiempo
1	10000000	11
2	20000000	13
3	30000000	14
4	40000000	19
5	50000000	24
6	60000000	29
7	70000000	31
8	80000000	68
9	90000000	46
10	100000000	44
11	110000000	47
12	120000000	56
13	130000000	54
14	140000000	61
15	150000000	61
16	160000000	63
17	170000000	66
18	180000000	71
19	190000000	74
20	200000000	78



2.8 Análisis de los resultados

De acuerdo a los resultados obtenidos en la gráfica se puede concluir que la notación asintótica se relaciona con la gráfica, puesto que ambas tienen una ecuación lineal, la cual platea la complejidad del ejercicio. Por esto se puede observar que a la hora compilar un programa con altos índices de datos el tiempo variaría de manera constante.

3. Tabla de multiplicar

3.1 Código en Word

```
public static void tablasMultiplicar(int n)
   {
      for(int i=1; i <= n; i++){
         for(int j=1; j <= n; j++){
            System.out.println(i+" * "+j+" = "+(i*j));
         }
         System.out.println();
      }
   }
```

3.2 Tamaño del problema (llamado también “n”)

El tamaño del problema son el número de iteraciones que realizo en el ciclo.

3.3 Cuánto se demora cada línea

```
public static void tablasMultiplicar(int n)
   {
      for(int i=1; i <= n; i++){ //C_1*(n+1) + C_2
         for(int j=1; j <= n; j++){ // (C_3*(n+1) + C_4)*n
            System.out.println(i+" * "+j+" = "+(i*j)); //C_5*n*n
```

```
            }
            System.out.println(); //C_6*n
        }
    }
```

3.4 Ecuación

$T(n) = C_1*(n+1) + C_2 + C_3*(n+1)*n + C_4*n + C_5*n*n + C_6*n$

3.5 Notación O dela solución de la ecuación

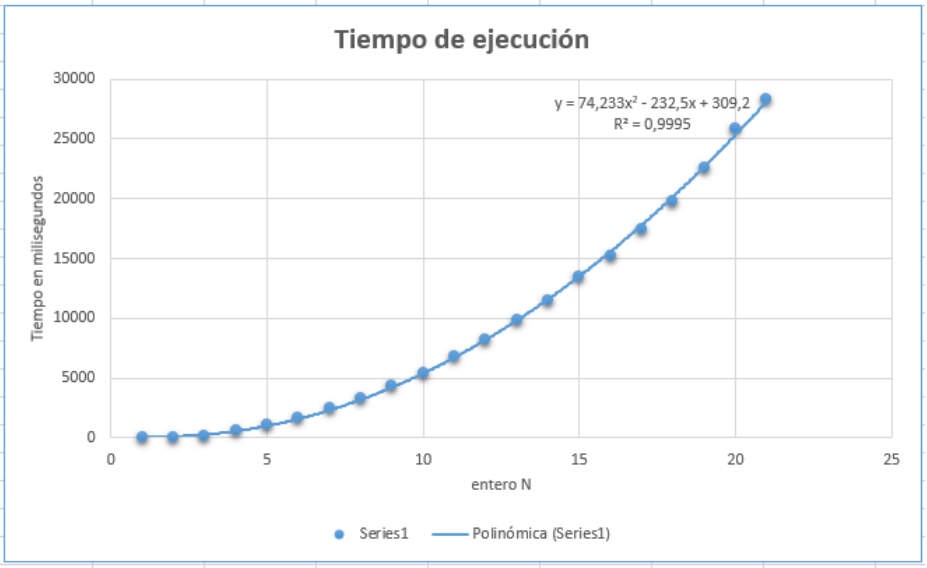
T(n) es O(C_1*(n+1) + C_2 + C_3*(n+1)*n + C_4*n+ C_5*n*n + C_6*n), por definición de O
T(n) es O(C_3*(n+1)*n + C_5*n*n) Regla de la suma
T(n) es O(C_5*n*n) Regla de la suma
T(n) es $O(n^2)$ Regla del producto

3.6 Explicado en palabras

La complejidad asintótica (es decir, para valores grandes de n) para el peor de los casos (es decir, en el que el algoritmo hace más operaciones) para el algoritmo de las tablas de multiplicar desde 1 hasta n mediante ciclos es de $O(n^2)$.

3.7 Grafica

n	Milisegundos
10000	44
20000	164
30000	609
40000	1098
50000	1689
60000	2461
70000	3368
80000	4333
90000	5474
100000	6883
110000	8201
120000	9888
130000	11555
140000	13431
150000	15234
160000	17427
170000	19832
180000	22659
190000	25842
200000	28381



3.8 Análisis de los resultados

De acuerdo a los resultados obtenidos en la gráfica se puede concluir que la notación asintótica se relaciona con la gráfica, puesto que ambas tienen una ecuación parabólica y creciente, la cual platea la complejidad del ejercicio. Por esto se puede observar que a la hora compilar un programa con altos índices de datos el tiempo variaría de manera creciente.