

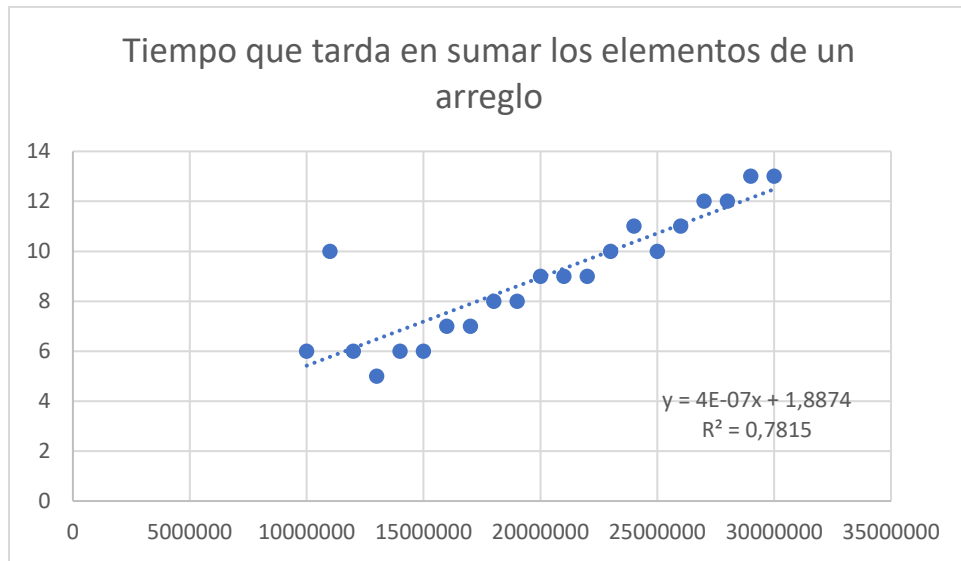
Taller 05

1. Insertion sort.



- De los resultados podemos concluir que a medida que aumenta el volumen de los datos incrementa el tiempo en ordenarlos. Además, podemos ver que los datos se ajustan casi perfectamente a un modelo lineal a pesar de que su notación O es $O(n^2)$. Sin embargo, tiene un R^2 de 0.99, por lo tanto, el error debe estar en que son muy pocos datos como para sacar conclusiones de su tendencia.
- No, no es un algoritmo apropiado para ordenar volúmenes de datos muy grandes pues se tarda demasiado tiempo en hacerlo. Por ejemplo, para valores entre 10 y 20 millones fue imposible calcular el tiempo que se tardaba en ordenarlos. Es más eficiente cuando se tarda de volúmenes pequeños, pues son menos datos que debe de comparar entre si para determinar el orden correcto.
- Su complejidad es:
 $T(n)$ es $O(n^2)$

2. Sumar los elementos de un arreglo.



- De los datos podemos percibir que hay uno de ellos muy alejado de la línea de tendencia que impiden que R^2 este tan cerca al uno como lo desearíamos. Sin embargo, este se ajusta a un modelo lineal con un coeficiente de regresión de 0.78, lo cual esta bien pues la mayoría de estos no pasan por la recta de tendencia.
- La diferencia significativa entre el tiempo de ejecución entre la implementación con recursión y la implementación con ciclos es que al hacerlo con ciclos permite calcular la suma de volúmenes más grandes de números, mientras que al hacerlo con recursión no es tan eficiente y es imposible calcularlo para tamaños tan grandes.
- La complejidad asintótica para el peor de los casos es:

$$T(n) = c1 + c2 + c3(n+1) + c4n + c5$$

$T(n)$ es $O(c3n)$ Regla de la suma

$T(n)$ es $O(n)$ Regla del producto