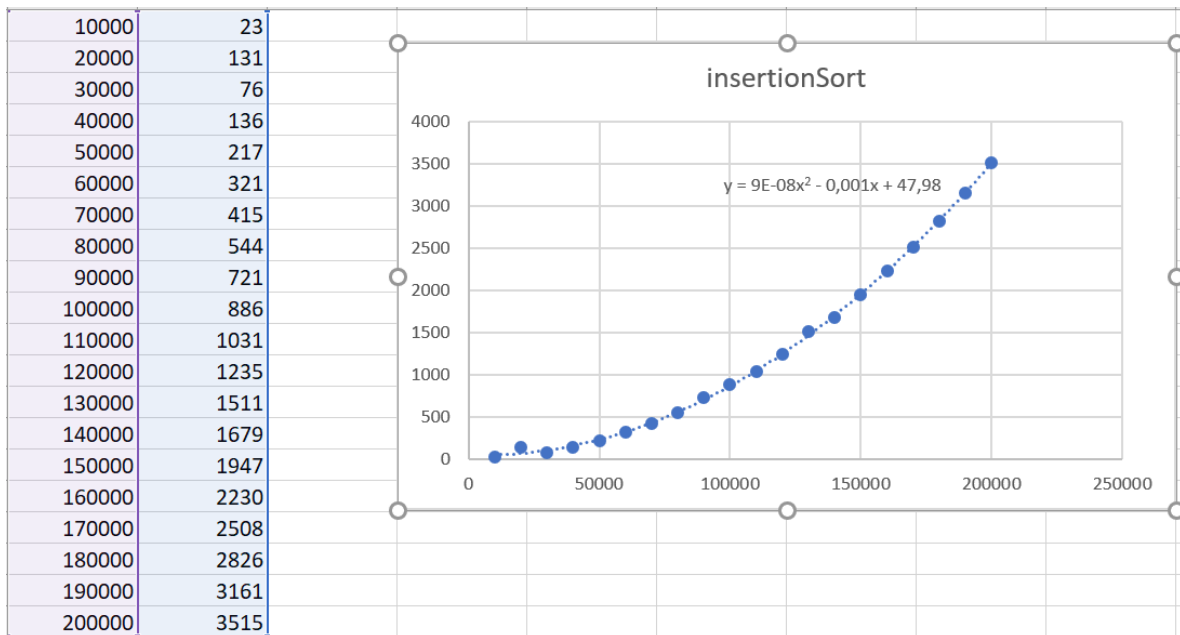


Taller 5: Complejidad de Algoritmos

Sara Rodriguez y Stiven Yepes

Punto 1: insertionSort

$$T(n) = O(n^2)$$



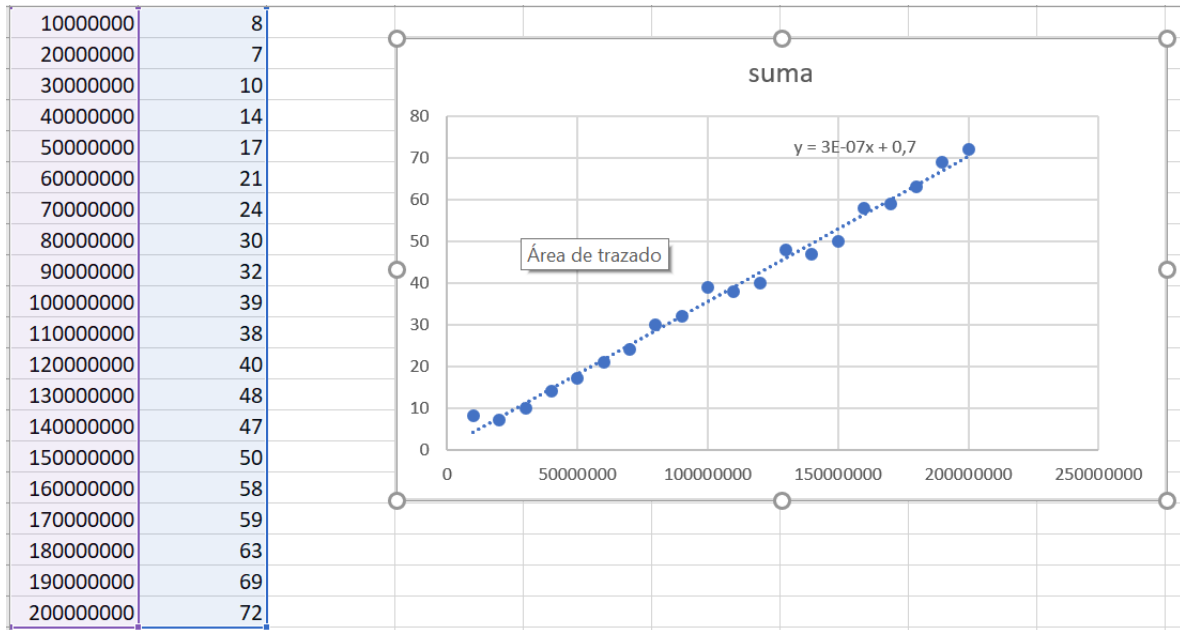
Como Podemos ver en el gráfico, la complejidad de este algoritmo es exponencial de orden n^2 . Al no ser tan altos los datos, podría confundirse la gráfica con una tendencia lineal, pues la curvatura no es tan notoria, pero al analizar el código vemos que afectivamente es una complejidad exponencial.

¿este algoritmo es apropiado para ordenar grandes volúmenes de datos?

Consideramos que sí, pues la complejidad de este algoritmo no es tan alta como lo podrían ser otros de orden n^3 o 2^n , así que, si bien se demora con datos muy altos, al final no es un tiempo exagerado.

Punto 2: suma

$$T(n) = O(n)$$



Como podemos ver en la gráfica, la complejidad de este algoritmo es lineal, de orden n , cuya pendiente está dada por la formula $y=3E-07x + 0,7$. se observa una pequeña distorsión en algunos de los datos, pero aun así siguen la misma tendencia.

¿existe una diferencia significativa con respecto a la implementación con recursión?

Si, la implementación con recursión tarda considerablemente más. En nuestra experimentación, el método recursivo no suporto el mismo valor de los datos, pues marcaba un error de desbordamiento de pila, solo se pudo hacer con valores mas pequeños.