

Universidad de Sonora
Lic. Ciencias de la Computación

Tarea 1: Introducción

- Análisis de Algoritmos -

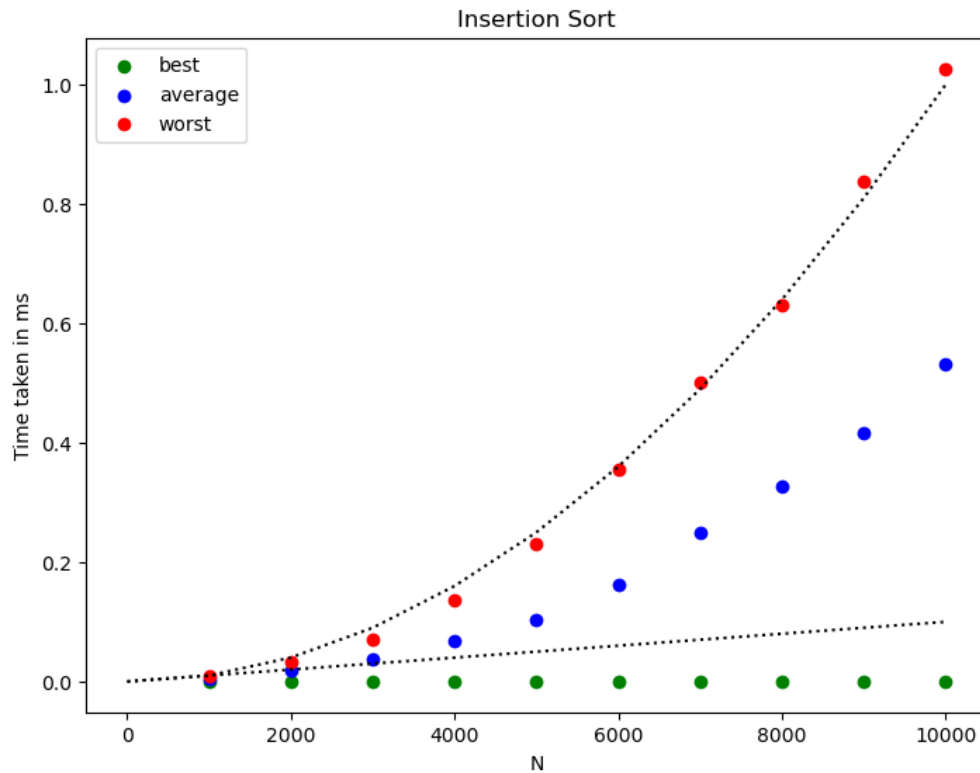
Enrique Alejandro Giottonini Herrera
21 de febrero de 2023

1.

Codificar el algoritmo de ordenamiento por inserción (**INSERTION-SORT**), y calcular el tiempo de ejecución para diez valores distintos de n . Realizar lo anterior para un arreglo ordenado en orden creciente (mejor caso), ordenado en forma decreciente (peor caso) y un arreglo aleatorio (caso promedio). Para cada caso calcular los tiempos y graficarlos.

$$N = k \times 1000$$

k	Time in ms.		
	Best	Average	Worst
1	3.9e-05	0.004437	0.0089
2	4.9e-05	0.018812	0.033794
3	9.4e-05	0.038454	0.071148
4	0.000121	0.068145	0.136728
5	0.000167	0.104135	0.230272
6	0.00015	0.161154	0.355639
7	0.00021	0.250216	0.500821
8	0.000226	0.326638	0.631154
9	0.000241	0.416636	0.836822
10	0.000354	0.531528	1.027353



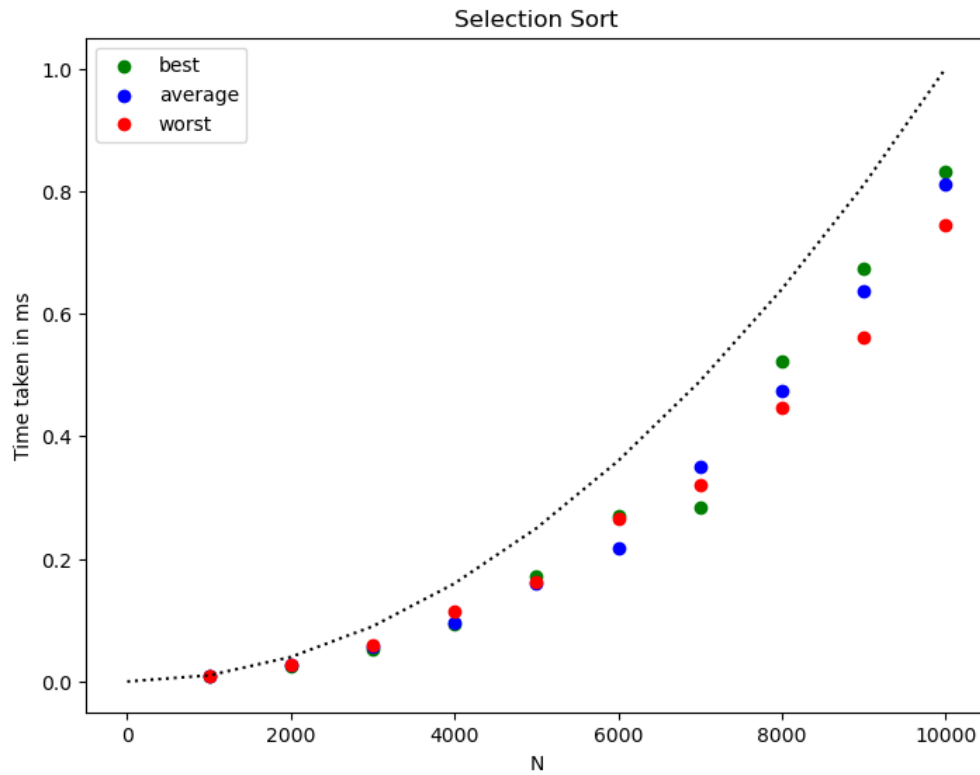
Observamos un comportamiento $O(n^2)$ para el caso promedio y peor. $O(n)$ para el mejor caso.

2.

Codificar el algoritmo de ordenamiento por selección (**SELECTION-SORT**) y calcular los tiempo de cómputo para diez valores distintos de n . Realizar lo anterior para un arreglo ordenado en orden creciente (mejor caso), ordenado en forma decreciente (peor caso) y un arreglo aleatorio (caso promedio). Para cada caso calcular los tiempos y graficarlos.

$$N = k \times 1000$$

Time in ms.			
k	Best	Average	Worst
1	0.007824	0.008779	0.007881
2	0.025799	0.026779	0.028041
3	0.051939	0.057631	0.05977
4	0.09482	0.09638	0.114162
5	0.170602	0.159569	0.163073
6	0.27077	0.21724	0.264626
7	0.284462	0.351367	0.320712
8	0.522975	0.4731	0.446152
9	0.67342	0.637975	0.560119
10	0.830803	0.811207	0.743811



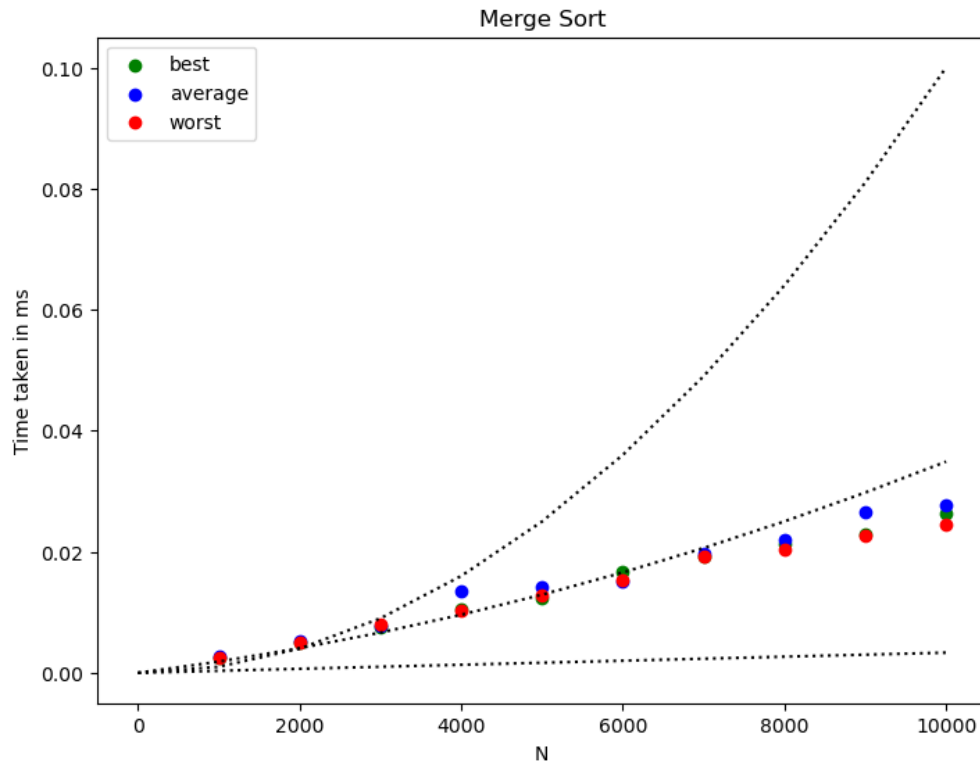
Observamos un comportamiento $O(n^2)$ para todos los casos.

3.

Codifique el algoritmo de ordenamiento por confluencia (**MERGE-SORT**) y calcular el tiempo de ejecución para diez valores distintos de n . Realizar lo anterior para un arreglo ordenado en orden creciente (mejor caso), ordenado en forma decreciente (peor caso) y un arreglo aleatorio (caso promedio). Para cada caso calcular los tiempos y graficarlos.

$$N = k \times 1000$$

k	Time in ms.		
	Best	Average	Worst
1	0.002496	0.002659	0.002419
2	0.00493	0.005136	0.004925
3	0.007453	0.007878	0.008016
4	0.010541	0.013392	0.010229
5	0.012291	0.014277	0.012893
6	0.016619	0.015182	0.015237
7	0.019261	0.0198	0.019167
8	0.021358	0.021905	0.020427
9	0.022903	0.026572	0.022608
10	0.02625	0.027609	0.024473



Observamos un comportamiento $O(n \log n)$ mayor a $O(n)$ pero menor que $O(n^2)$

4.

Determinar la función $T(n)$, en el mejor y el peor de los casos, del algoritmo de ordenamiento por selección.

Algorithm 1 Selection-Sort (Mejor Caso: Lista Ordenada)

1: for $i = 1$ to $n - 1$	c_1	$t = n$
2: $\min j = i$	c_2	$t = n - 1$
3: $\min x = A[i]$	c_3	$t = n - 1$
4: for $j = i + 1$ to n	c_4	$t = (n - 1)^2$
5: if $A[j] < \min x$ then	c_5	$t = (n - 1)^2 - 1$
6: $\min j = j$	c_6	$t = 0$
7: $\min x = A[j]$	c_7	$t = 0$
8: end if		
9: end for		
10: $A[\min j] = A[i]$	c_8	$t = n - 1$
11: $A[i] = \min x$	c_9	$t = n - 1$
12: end for		

$$T(n) = c_1(n) + c_2(n - 1) + c_3(n - 1) + c_4(n - 1)^2 + c_5((n - 1)^2 - 1) + c_8(n - 1) + c_9(n - 1)$$

$$T(n) = (c_4 + c_5)n^2 + (c_1 + c_2 + c_3 - 2c_4 - 2c_5 + c_8 + c_9)n - (c_2 + c_3 + c_4 + c_8 + c_9)$$

$$T(n) = An^2 + Bn + C$$

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

Algorithm 2 Selection-Sort (Peor Caso: Lista Ordenada al revés)

1: for $i = 1$ to $n - 1$	c_1	$t = n$
2: $\min j = i$	c_2	$t = n - 1$
3: $\min x = A[i]$	c_3	$t = n - 1$
4: for $j = i + 1$ to n	c_4	$t = (n - 1)^2$
5: if $A[j] < \min x$ then	c_5	$t = (n - 1)^2 + n$
6: $\min j = j$	c_6	$t = (n - 1)^2 + n - 1$
7: $\min x = A[j]$	c_7	$t = (n - 1)^2 + n - 1$
8: end if		
9: end for		
10: $A[\min j] = A[i]$	c_8	$t = n - 1$
11: $A[i] = \min x$	c_9	$t = n - 1$
12: end for		

$$T(n) = c_1(n) + c_2(n - 1) + c_3(n - 1) + c_4(n - 1)^2 + c_5((n - 1)^2 + n)$$

$$+ c_6((n - 1)^2 + n - 1) + c_7((n - 1)^2 + n - 1) + c_8(n - 1) + c_9(n - 1)$$

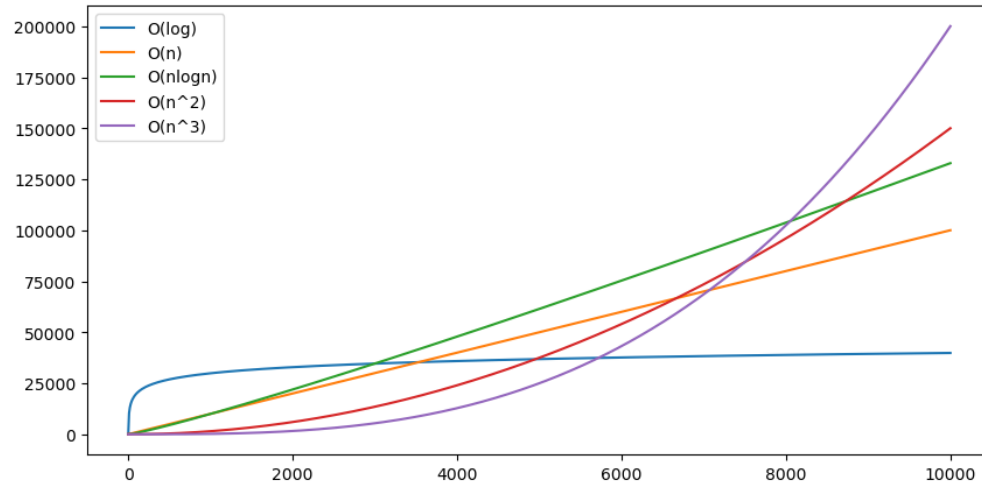
$$T(n) = (c_4 + c_5 + c_6 + c_7)n^2 + (c_1 + c_2 + c_3 - 2c_4 - c_5 - c_6 + c_7 + c_8 + c_9)n - (c_2 + c_3 + c_4 + c_8 + c_9)$$

$$T(n) = An^2 + Bn + C$$

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

5.

Realizar la graficación de las siguientes funciones: $f(n) = \log(n)$, $f(n) = n$, $f(n) = n \log(n)$, $f(n) = n^2$, $f(n) = n^3$. Todas las curvas deben ser colocadas en la misma gráfica.



Se observa que para ciertos N se cumpla que algún algoritmo sea mejor que otro, pero cuando $N \rightarrow \infty$ esta condición cambia.