UNIVERSIDAD DE COSTA RICA

Análisis de Algoritmos y Estructuras de Datos

CI-0116

Tarea 1

II Parte

ELABORADO POR

Josué Retana Rodríguez - C06440

PROFESOR ASESOR

Dr. Arturo Camacho Lozano

2021

Introducción

En este trabajo se realizará un análisis de ciertos algoritmos de ordenamiento vistos en el curso CI-0116. Dichos algoritmos consistirán en: ordenamiento por selección, ordenamiento inserción y ordenamiento por mezcla, ordenamiento por montículos, ordenamiento rápido y ordenamiento por residuos; los algoritmos serán implementados en el lenguaje de programación C++.

Se crearán arreglos al azar de tamaños 50 000, 100 000, 150 000 y 200 000 y se ejecutarán 3 veces cada algoritmo, esto con el fin de analizar su eficiencia y realizar comparaciones. Además, el análisis se referirá a temas como la variación de los tiempos en las tres corridas y si su forma graficada es la esperada.

Es importante mencionar que todos los resultados se agregarán a una tabla resumen que incluirá los promedios de las tres ejecuciones. También se realizarán gráficos de los tiempos promedio contra el tamaño del arreglo para cada algoritmo y se realizarán conclusiones a partir de estos.

Tiempos de Ejecución de los Algoritmos en ms

Análisis tiempo-tamaño de Algoritmos					
Algoritmo	***Tamaño***	***Ejecución 1***	***Ejecución 2***	***Ejecución 3***	***Promedio***
Selección	50 000	1399	1392	1374	1388.333333
	100 000	5364	5374	5434	5390.666667
	150 000	17543	16387	17232	17054
	200 000	32276	31825	31920	32007
Inserción	50 000	908	922	948	926
	100 000	3607	3599	3600	3602
	150 000	9562	9490	9625	9559
	200 000	19536	20242	19315	19697.66667
Mezcla	50 000	12	11	12	11.66666667
	100 000	24	23	24	23.66666667
	150 000	35	36	35	35.33333333
	200 000	45	45	47	45.66666667
Heapsort	50 000	8	7	7	7.333333333
	100 000	14	15	16	15
	150 000	22	23	23	22.66666667
	200 000	31	32	32	31.66666667
Quicksort	50 000	3	2	3	2.666666667
	100 000	5	5	6	5.333333333
	150 000	7	8	8	7.666666667
	200 000	11	12	11	11.33333333
Radixsort	50 000	2	2	2	2
	100 000	4	4	5	4.333333333
	150 000	6	7	6	6.333333333
	200 000	9	10	10	9.666666667

Con respecto a las ejecuciones del algoritmo de ***selección***: comparando las tres corridas con un mismo tamaño, en los cuatro casos se puede notar que realmente duración fue muy similar. Sin embargo, cuando se comparan los tiempos promedios se puede notar una gran diferencia, por ejemplo, la duración teniendo 100 000 valores supera por casi cuatro veces la duración si se tienen 50 000 valores, es decir la mitad.

Con respecto a las ejecuciones del algoritmo de ***inserción***: comparando las tres corridas con un mismo tamaño, en los cuatro casos se puede notar que no hubo una variación grande entre los tiempos. Sin embargo, cuando se comparan los tiempos promedios se puede notar una diferencia mediana, aunque aún con una gran tendencia a durar más, en este caso, la duración teniendo 200 000 valores supera por aproximadamente cinco veces la duración si se tienen 100 000 valores, es decir la mitad.

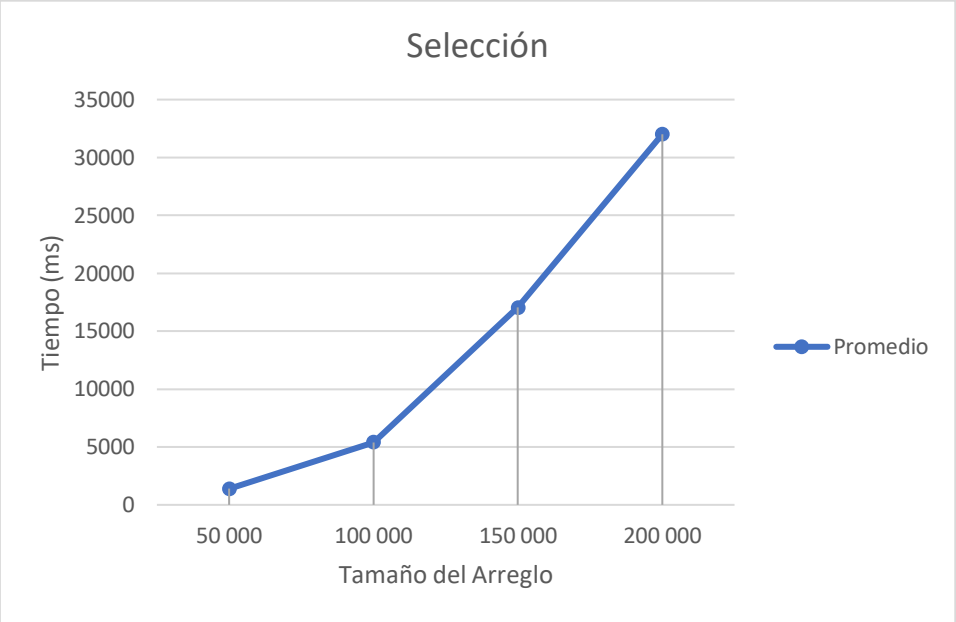
Con respecto a las ejecuciones del algoritmo de ***mezcla***: comparando las tres corridas con un mismo tamaño, en los cuatro casos se puede notar que es mínima e inclusive en un caso de repitió el tiempo de duración. Si se comparan los tiempos promedios se puede notar una diferencia mínima y notablemente superior respecto a los algoritmos de selección e inserción.

Con respecto a las ejecuciones del algoritmo de ***heapsort***: en las tres ejecuciones y en los tres tamaños, el común denominador es la falta de variación de tiempo por lo tanto el algoritmo denota estabilidad. A su vez, se según aumenta el número de elementos por ordenar, se nota un aumento constante del tiempo.

Con respecto a las ejecuciones del algoritmo de ***quicksort***: el algoritmo presentó grandes resultados, y bastante superiores a los algoritmos previos analizados. No hay gran variación entre los tiempos, y entre los distintos tamaños existe un aumento gradual de la duración, sin embargo, siendo muy veloz inclusive en sus peores casos.

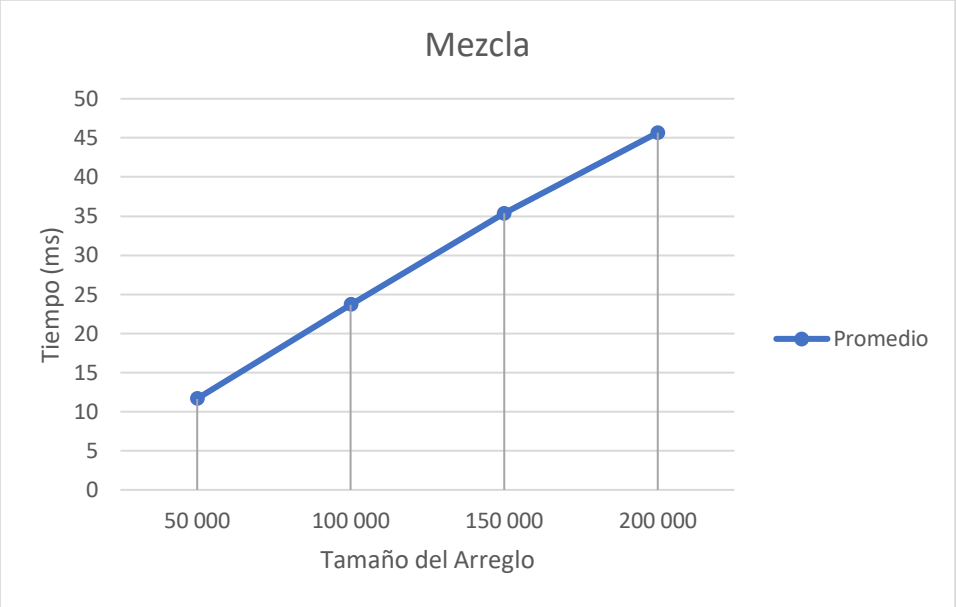
Con respecto a las ejecuciones del algoritmo de ***radixsort:*** este algoritmo tiene resultados notablemente superiores a los demás, el más cercano en cuestión de duración sería el quicksort, sin embargo, sigue siendo superior el ordenamiento por residuos. Entre los tiempos existe una variación mínima del tiempo de duración, al igual que en los tamaños.

Gráficos de Algoritmos con tiempos promedio contra el tamaño del arreglo

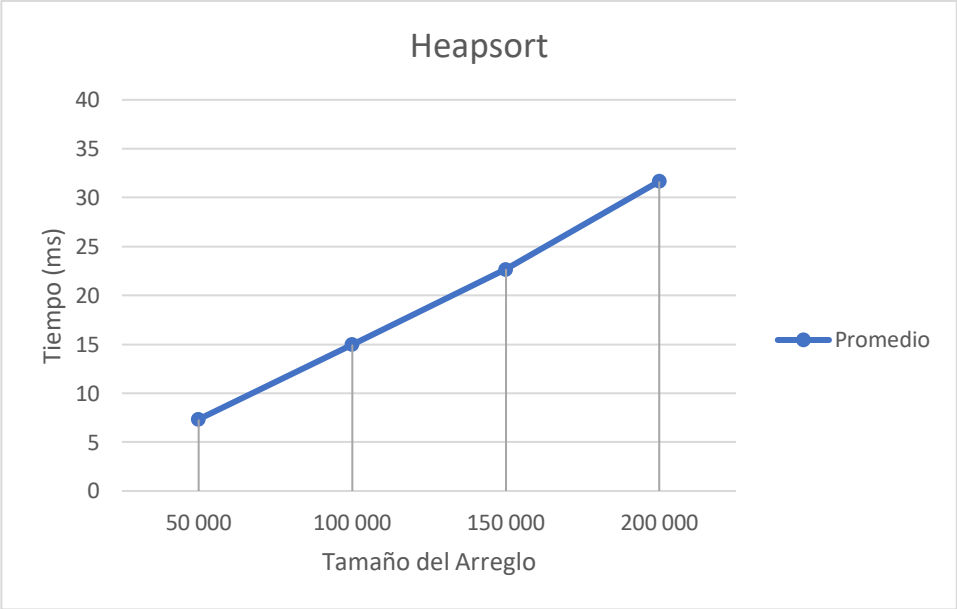


Este algoritmo con tiempo de complejidad Θ(n2) se espera que tenga una curva similar a una parábola. La forma de la curva es la esperada.

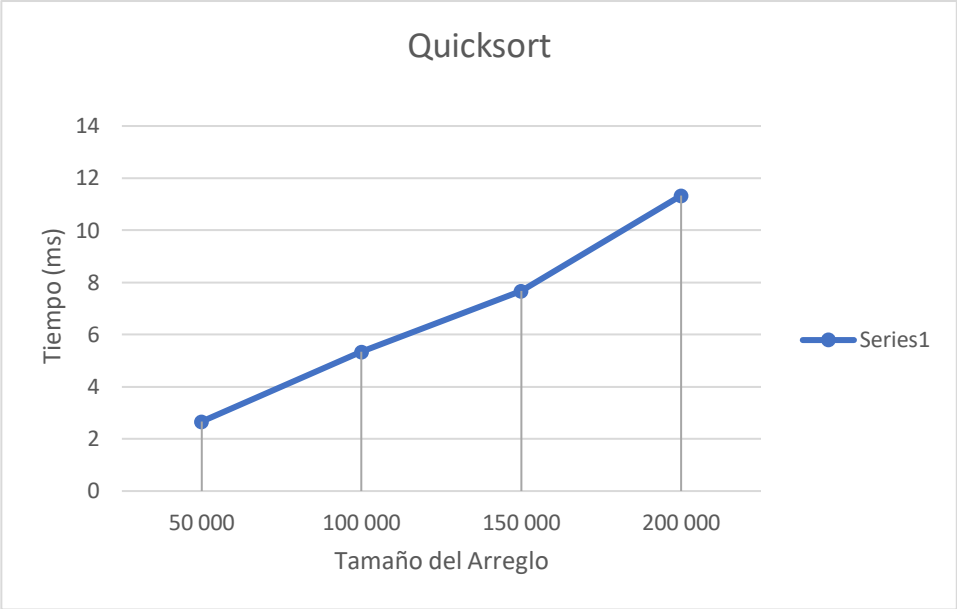
Este algoritmo con tiempo de complejidad Θ(n2) se espera que tenga una curva similar a una parábola. La forma de la curva es la esperada.



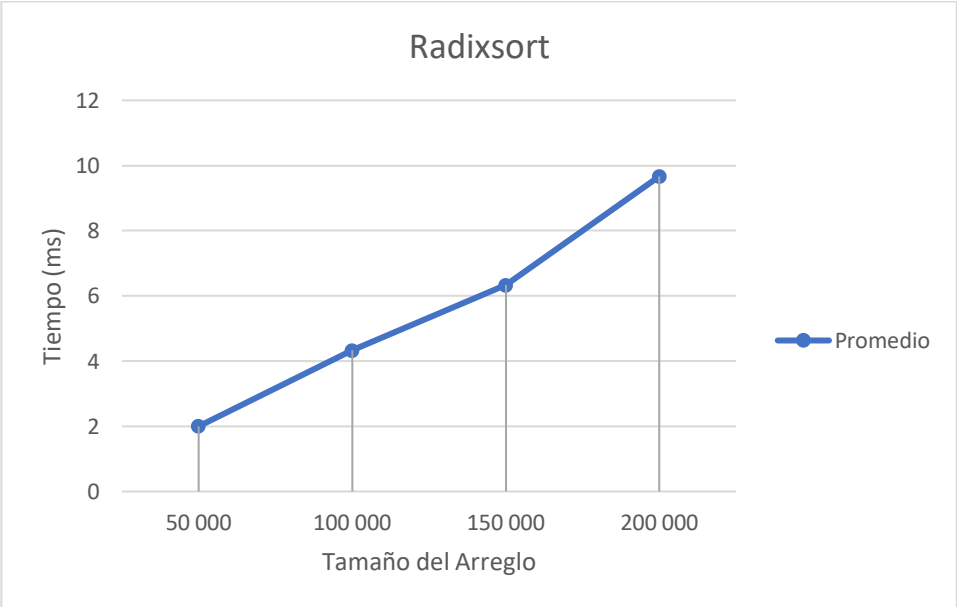
Este algoritmo con tiempo de complejidad Θ(n log n) se espera que tenga una curva aproximadamente lineal. La forma de la curva es la esperada.



El algoritmo por montículos tiene un tiempo de complejidad Θ(n log n), por lo tanto se espera que tenga una curva aproximadamente lineal y en este caso cumple con dicha característica.

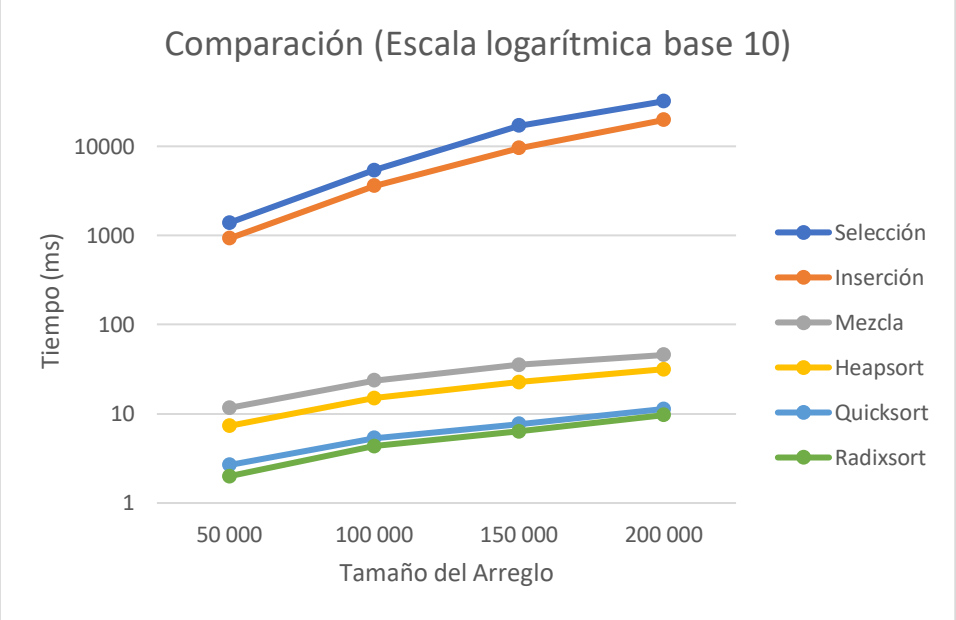


El ordenamiento rápido tiene un tiempo de complejidad de Θ(n2) en la práctica, pero en promedio el tiempo es de Θ(n log n), por lo tanto esperaría una curva lineal y en este caso se cumple.



En el caso de ordenamiento por residuos se tiene un tiempo de complejidad de Θ(n2), por lo tanto, su curva es aproximadamente lineal y esto se cumple en las pruebas.

Comparación de Tiempos Promedios



La relación entre curvas de cada algoritmo es la esperada. En el caso de selección e inserción con tiempos de complejidad iguales las curvas son similares en forma y están ligeramente tornadas parabólicamente. En el caso de mezcla, montículos, rápido y residuos se mantiene una forma aproximadamente lineal en todos, cumpliendo así lo que dicta la teoría. En general, el algoritmo por residuos presenta una superioridad, aunque muy ajustada con el ordenamiento rápido, en cuestión de tiempo, mientras que selección es el más lento de todos.

Conclusión

A partir de los resultados del análisis se comprueba que todos los algoritmos cumplieron con lo que establece la teoría. En términos de rapidez el ganador es el algoritmo de ordenamiento por residuos, aunque muy de cerca le seguían el ordenamiento rápido seguido de ordenamiento por montículos. Los algoritmos mencionados anteriormente, junto con ordenamiento por mezcla, muestran una gran superioridad en comparación a los algoritmos de inserción y selección. El algoritmo más lento de todos corresponde al de selección, donde toma inclusive medio minuto ordenando arreglos de tamaño 200 000.