Implementación de Tries

Ilse Córdova Sánchez 181901

Estructuras de datos avanzadas

Instituto Tecnológico Autónomo de México

19 de noviembre 2019

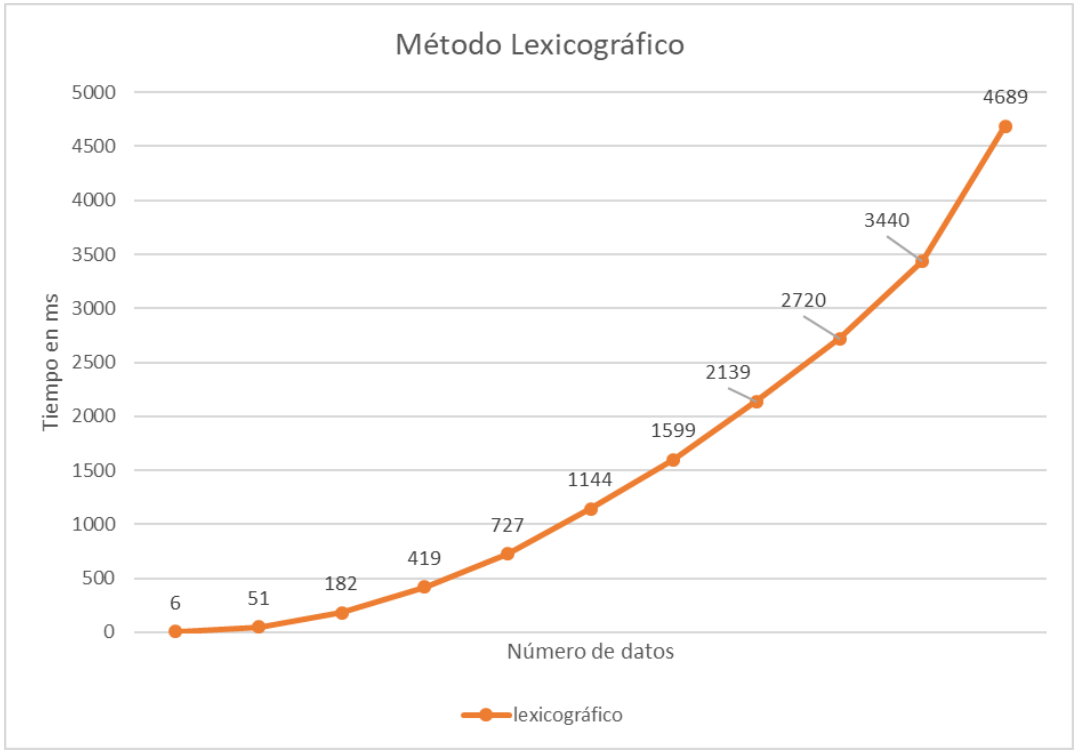
I. Comparación de datos

Nota: El tiempo se midió en ms.

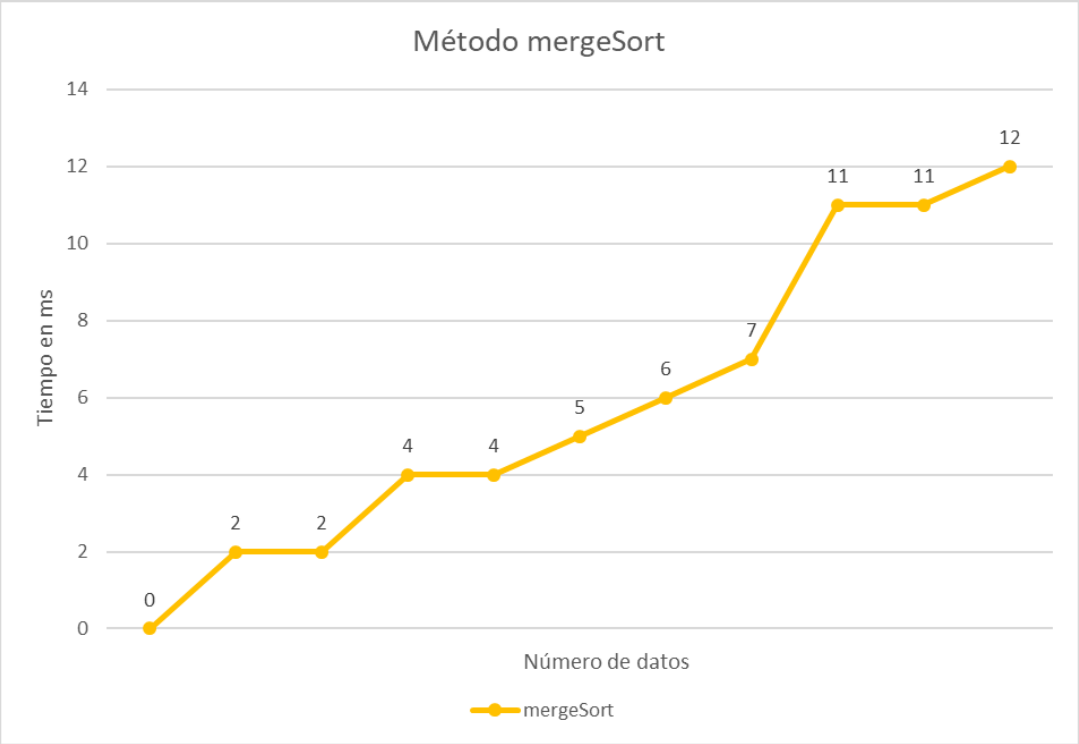
Número de datos	Método de ordenamiento lexicográfico	Método de ordenamiento mergeSort
1,000	6.0 ms	0.0 ms
5,000	51.0 ms	2.0 ms
10,000	182.0 ms	2.0 ms
15,000	419.0 ms	4.0 ms
20,000	727.0 ms	4.0 ms
25,000	1144.0 ms	5.0 ms
30,000	1599.0 ms	6.0 ms
35,000	2139.0 ms	7.0 ms
40,000	2720.0 ms	11.0 ms
45,000	3440.0 ms	11.0 ms
50,000	4689.0 ms	12.0 ms

II. Gráficos

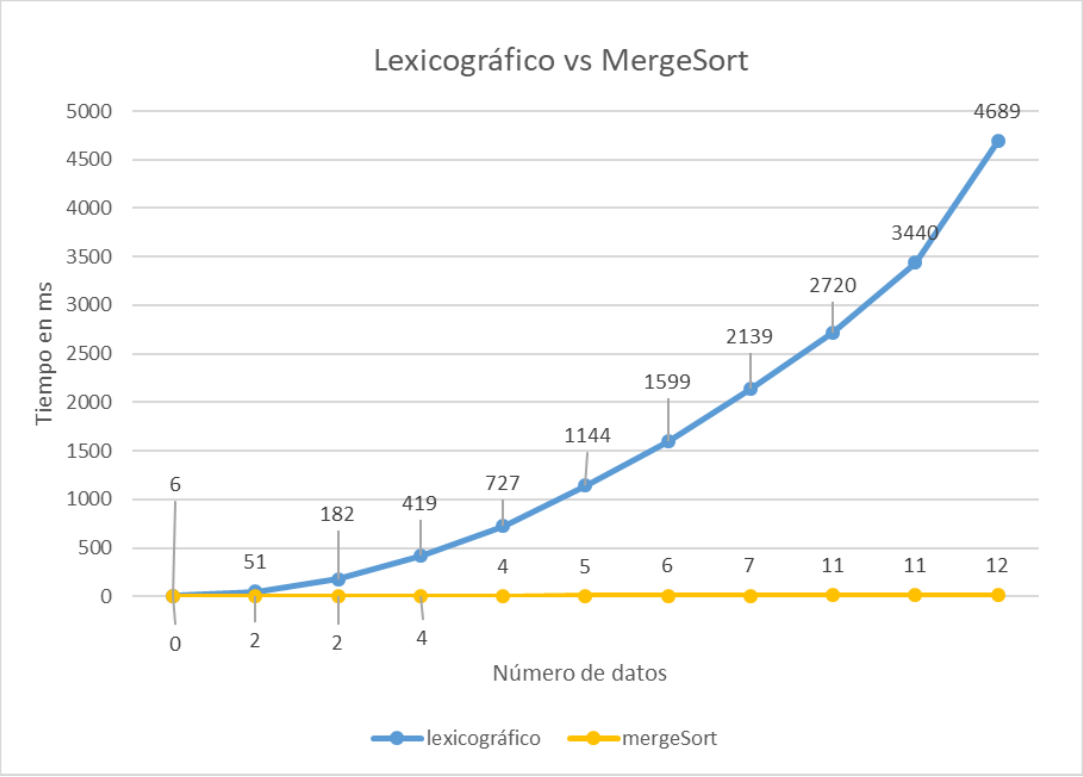
Ordenamiento Lexicográfico



Ordenamiento MergeSort



Comparación de métodos



III. Análisis

El experimento realizado constó en programar un método de ordenamiento lexicográfico propio de la estructura de un Trie para posteriormente insertar diferentes cantidades de datos y registrar el tiempo que le llevaba ejecutar el método anteriormente mencionado. Una vez que se insertaron mínimo 50,000 datos y se llevaron a cabo las pruebas pertinentes con el método de ordenamiento lexicográfico, se procedió a utilizar el algoritmo de ordenamiento mergeSort para repetir el experimento.

Después de realizar las pruebas pertinentes y mostrar los datos obtenidos de estas en forma de tablas y gráficos, se puede llegar a varias conclusiones. La primero que puede observarse es que, bajo cualquier circunstancia, el algoritmo mergeSort es el que menos tiempo le lleva ordenar los datos introducidos sin importar el número de palabras que se tengan que analizar; su comportamiento asintótico es de n log(n).

Por otra parte, el método de ordenamiento lexicográfico llegó a completar su ejecución en 4689 ms al tener que ordenar 50,000 datos mientras que al algoritmo mergeSort solamente le llevó 12 segundos. El tiempo que le lleva al método lexicográfico ordenar las palabras aumenta rápidamente conforme se van manejando más datos. En contraste, en el caso del algoritmo mergeSort el tiempo se mantiene constante al manejar cierto número de datos (le llevó 4.0 ms tanto ordenar 15,000 como 20,000 datos).

Es cierto que podrían utilizarse otros algoritmos de ordenamiento y compararlos con el método de ordenamiento lexicográfico; sin embargo no parece viable la posibilidad de que dicho método logre ser más eficiente que el algoritmo mergeSort, esto debido a que la complejidad de la estructura aumenta con cada nivel de profundidad ya que cada nodo tiene mínimo 26 hijos (considerando el caso en el que se utilice el alfabeto). Tal vez si las palabras se insertaran en un mergeTree entonces un método de ordenamiento lexicográfico propio de esta estructura podría llegar a ser más eficiente que al algoritmo mergeSort.