Árboles Trie

Fonseca Zárate Israel 183708

Se comprobó la eficiencia de la estructura de datos llamada árboles trie, para ordenar conjuntos de palabras, comparándolo con el algoritmo de ordenamiento merge sort, el cual tiene una eficiencia de O(n log n), siendo de los algoritmos más eficientes de su tipo.

Para ello se implementó un método que recorre el árbol de manera recursiva, de manera similar al recorrido pre-orden, pues visita primero la raíz, y va pasando por todos los hijos del menor al mayor. Conforme los recorre, va formando la palabra, la cual es guardada en una lista al comprobar que sea fin de palabra.

Como primeras diferencias, el algoritmo merge sort requiere que se le pase un arreglo con las palabras a ordenar, por otro lado el árbol trie, requiere insertar de una por una las n palabras, es en el proceso de inserción donde son ordenadas en la estructura, a continuación se debe recorrer la estructura, tomando todas las palabras y las va sobrescribiendo sobre el arreglo, de manera similar a merge sort que regresa el mismo arreglo pero ordenado.

Para los experimentos se probó con 1000 palabras, luego con 5000 y a partir de ahí se fue aumentando de 5 mil en 5 mil palabras, hasta llegar a 50 mil, en cada uno se midió el tiempo de ejecución obtenido por el algoritmo merge sort y por el árbol trie.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Tiempo de ejecución | | milisegundos | |  |  |  |  |  |  |  |
| #Palabras | 1000 | 5000 | 10000 | 15000 | 20000 | 25000 | 30000 | 35000 | 40000 | 45000 | 50000 |
| MergeSort | 4.5792 | 19.4531 | 24.9126 | 37.4316 | 46.1528 | 54.82 | 58.7293 | 76.0523 | 81.7741 | 99.2453 | 119.2791 |
| Árbol Trie | 12.9401 | 32.2472 | 57.8784 | 83.1701 | 86.4542 | 123.8654 | 132.9448 | 121.0809 | 111.5955 | 115.4321 | 118.5631 |

Desde un principio es más eficiente el algoritmo merge sort, solamente con 50 mil palabras llega a rebasar en tiempo de ejecución a los árboles trie, los cuales, a mayor número de palabras, tiende en cierto punto a mantenerse estable. Esto quizá es debido a que el árbol trie se va haciendo más frondoso y en parte también por el tamaño de las palabras, deja aumentar niveles, lo que lo va haciendo más eficiente, en este caso, el tiempo de ejecución deja de aumentar al llegar a las 30 mil palabras.

Esté método difícilmente es más eficiente que los algoritmos de ordenamiento, requeriría quizá que se trate de un gran número de palabras. Además, sus usos son más limitados, pues se basa en el uso de llaves, por lo general palabras u otros símbolos que fueron previamente definidos, y no de objetos comparables, de modo que no puede guardar cualquier tipo de dato, además por la implementación hecha en clase, solo puede guardar palabras.