Licenciatura en ciencia de la computación

Algoritmo de Strassen

Complejidad

Profesor:
Nicolas Thériault

Autor:
Sergio Salinas
Danilo Abellá

Contents

1 Introducción

2 Estrategias utilizadas

2.1 Arreglo ordenado con Mergesort

Esta estrategia consiste en crear un arreglo de dos dimensiones, la primera dimensión tiene largo 21000 que es una estimación de la cantidad de palabras que tiene al archivo y cada arreglo asociado a este tiene un largo de 30, que también es una estimación de la cantidad de letras que tiene cada palabra. por lo que en total este método gasta 63000 espacios de memoria cada vez que se ejecuta.

2.2 Árbol AVL

Esta estrategia consiste en usar un árbol binario balanceado conocido como Árbol AVL, para la implementación se le hizo una ligera modificación que es agregar a la estructura de un nodo la frecuencia que este aparece, de esta forma en vez de apilar los elementos repetidos en nodos, solo aumenta el contador de frecuencia de este logrando una notoria mejora en memoria, ya que el árbol implementado solo va a ocupar tantos nodos como **palabras únicas** haya en el archivo. De está forma la estructura del árbol es la que se muestra en la tabla 1.

Table 1: Estructura de un nodo de un árbol AVL

Nodo
Data
Altura
Frecuencia
Puntero Izquierdo
Puntero Derecho

2.3 Tabla Hash

Para implementar el algoritmo con funciones de hash se utilizar un arreglo de largo fijo para almacenar las cabeceras de las listas enlazadas que contienen las palabras que se desean almacenar, este arreglo en un principio solo almacena elementos nulos, cuando se quiere ingresar una palabra se pasa esta palabra por una función hash y luego el resultado es utilizado como incide y se agrega a la lista que corresponde a ese indice, cada elemento es ingresado al inicio de la lista.

La función hash que utiliza el algoritmo es el algoritmo djb2 [1], debido a que este permite mantener una buena distribución entre el largo del arreglo, logrando que en la mayoría de las veces las únicas colisiones que haya son de las palabras que son iguales.

La tabla hash siempre va a guardar todas las palabras que tenga el archivo, pero se puede hacer un intercambio de tiempo memoria en el largo del arreglo de punteros, entre más largo sea más memoria utilizara pero habrán menos colisiones lo que se traduce en menos tiempo en operaciones de busqueda (la búsqueda llega a ser de costo 1 en este caso) y conteo, entre más corto menos memoria utilizara pero se gasta más tiempo en hacer las operaciones de búsqueda y conteo dentro de las listas enlazadas. Para propósitos de la medición de tiempo se decidió que el largo del arreglo de punteros sea de largo 4000, esto tomando en cuenta que la cantidad de palabras únicas en el archivo es de 3219, aunque se puede modificar sin problemas cambiando el valor de MAX en el define.

[1] `http://www.cse.yorku.ca/~oz/hash.html`