|  |
| --- |
| Materia:algoritmos y estructura de datos |
| Parcial Practico Nº2 |
|  |
|  |
|  |
|  |

**11/06/2015**

Alumnos:

Herrera, Nicolás 35135164

Malano, Leandro 38883701

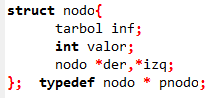
Quinteros, Nicolás 34351470

**DEfensa del algoritmo realizado**

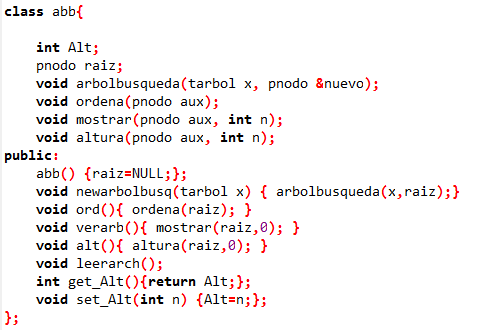
En este trabajo se desarrolló un programa que nos permite decir cuantas palabras se repiten y ordenarlas en un árbol de búsqueda en el cual hacemos que el programa lea un archivo de texto y también de la altura del árbol creado.

Se definimos la variable tipo string tarbol (typedef string tarbol;) de manera global.

Luego, creamos la estructura nodo esta formada por dos punteros (der e izq) y tenemos un string (inf) al ser un árbol binario.



Luego creamos la clase abb.



Donde en la parte en la parte privada de la función creamos el entero Alt, el nodo raíz, las funciones void arbolbusqueda(tarbol),void orderna(pnodo aux), mostrar(pnodo aux, int n) y altura(pnodo aux, int n).

El archivo se lee desde la función leerarch(), donde verifica si esta en buen estado y analiza los string que son enviados a la función newarbolbusq(tarbol x), donde justamente x es un string, a su vez esta función llama a la funcion arbolbusqueda(x,raiz), en donde recibe el string x y el nodo raíz, en esa función verifica si el nodo es NULL, si lo es crea el nodo nuevo que pasa a apuntar al string inf que almacena el valor de x, luego nuevo apunta al nodo der que tiene el valor del puntero nuevo que apunta a izq=NULL, luego compara si el string x es mayor o menor (alfabeticamente) al nodo nuevo que apunta a inf y si se da que el valor de x es igual al nodo que apunta a inf, nuevo apunta a valor que se incrementa de forma unitaria y se imprime el valor del nodo nuevo que apunta a valor ( muestra la cantidad de veces que se muestra una palabra) y el valor de x (palabras del todo el texto).

Luego en la función verarb(), se llama a la función mostrar(raiz,0) en donde ingresa el árbol generado por la función anterior, para ser visualizado de forma tal, en donde ingresa el nodo raíz no nulo, la cual es una función recursiva, y se envía como parámetro el mismo nodo que recibe como parámetro que apunta al nodo izq o der depende del caso y con un ciclo de repetición se muestra el ya ordenado.

Siguiendo, tenemos la función alt() , que llama a la función altura y que recibe como parámetro raíz y el 0. Se verifica si raíz es no nulo y y se llama recursivamente con el nodo apuntando a izq o der y sumando de n+1, si llega en momento que n es mayor a get\_Alt() (función que consigue la altura del arbol), entonces la función set\_Alt(int n), recibe como parámetro a n, que seria ya la altura del árbol.

Y por ultimo tenemos la función ord(), que llama a la función ordena(raiz), en donde recibe el nodo no nulo y de forma recursiva el nodo raíz va apuntando a izq o der para el acomodamiento de las palabras y ya acomodado, se imprimen por orden alfabético gracias a que la función muestra el nodo raíz que apunta a inf el cual almacena la palabra.

En el Main() se crea el árbol (abb) y se llaman las funciones para que esta muestren desde el programa principal.