**Análisis del algoritmo propuesto en la solución #3**

/\*

Mauricio Iván Cortés García

A00816689

Análisis y Diseño de Algoritmos

Prof. Roman Martinez

Tarea 2

Solución 3

SolucionTres.cpp

\*/

#include <iostream>

#include <fstream>

#include <time.h>

#include <windows.h>

#include <list>

using namespace std;

const int Posiciones\_Tabla\_Hash=1000;

/\*

Nombre: FuncionHash.

Entradas: Matrícula.

Salidas: Clave de 3 dígitos para introducir a tabla hash.

Función: Introduciendo una matrícula (int), se sacan los últimos 3 dígitos para obtener la clave para la tabla hash.

\*/

O(1)

int FuncionHash(int iDatos)

{

return iDatos % 1000;

}

/\*

Nombre: Encuentra\_Matricula.

Entradas: Lista de hash <int>, iDatos(int) - Matricula que se desea buscar.

Salidas: Booleano diciendo si la matricula existe o no en la tabla de hash

Función: Se busca en la lista dentro de la posicion de la tabla de hash para identificar la matriula a buscar.

\*/

bool Encuentra\_Matricula(list<int> Lista\_Hash[Posiciones\_Tabla\_Hash], int iDatos)

O(n)

{

list<int>::iterator Inicio=Lista\_Hash[FuncionHash(iDatos)].begin();

list<int>::iterator Final=Lista\_Hash[FuncionHash(iDatos)].end();

while(Inicio!=Final)

{

if(\*Inicio==iDatos)

return true;

++Inicio;

}

return false;

}

int main()

{

int Busca\_Mat, iDatos;

list<int> Lista\_Hash[Posiciones\_Tabla\_Hash];

ifstream Archivo\_Entrada;

ofstream Archivo\_Salida;

//Pregunta al usuario por la matrícula a buscar en el tabla de hash de matrículas y lo almacena en Busca\_Mat.

cout << "Que matricula deseas buscar?" << endl;

cin >> Busca\_Mat;

clock\_t startTime = clock();

Archivo\_Entrada.open("Archivo\_Matriculas.txt");//Abre el archivo de entrada.

//Mientras que el archivo no se acabe, se obtienen los datos e introducen en la tabla de hash

while(!Archivo\_Entrada.eof())

O(n)

{

Archivo\_Entrada >> iDatos;

Lista\_Hash[FuncionHash(iDatos)].push\_back(iDatos);

}

//Se despliega si la matricula buscada pertenece a la tabla(si es EXATEC o no).

Encuentra\_Matricula(Lista\_Hash,Busca\_Mat) ? cout << "Si es EXATEC." << endl : cout << "No es EXATEC." << endl;

//Despliega en un archivo .txt el tiempo de ejecucion del programa.

Archivo\_Salida.open("Tiempo de ejecucion Solucion Tres.txt");

Archivo\_Salida << double(clock()-startTime)/(double)CLOCKS\_PER\_SEC << " segundos." << endl;

Archivo\_Salida.close();

system("PAUSE");

return 0;

}

Orden de complejidad de tiempo: O(n)

Orden de complejidad de espacio: O(n)

Tipo de algoritmo: No Every-case