|  |  |  |
| --- | --- | --- |
| **CARRERA:**  Ingeniería de Software | **GUÍA**  No. 01 | **TIEMPO ESTIMADO:**  1 mes |
| **ASIGNATURA:**  Estructura de datos  NRC: 2967 | **FECHA DE ELABORACION:** 23-10-2019  **SEMESTRE**: septiembre 2019 – febrero 2020 | |
| **TÍTULO:**  Backtracking | **DOCENTE:** Ing. Fernando Solis | |

**OBJETIVO**

Comprender el significado y uso de Backtracking para la generación de códigos o algoritmos más eficientes.

**INSTRUCCIONES**

**Backtracking**

La técnica de backtracking está muy asociada a la recursividad, o más propiamente, a la estructura recursiva de la mayoría de los tipos de datos: listas, árboles, etc.

Esta técnica consiste básicamente en:

* Enumerar sistemáticamente las alternativas que existen en cada momento para dar con la solución a un problema.
* Se prueba una alternativa, guardando memoria del resto de alternativas.
* Si no damos con la solución, podemos dar marcha atrás (backtracking) y probar otra alternativa.

Esta técnica tan trivial resulta especialmente útil para los problemas de búsqueda en estructuras de datos. Especialmente si no existe a priori ningún criterio de búsqueda mejor o peor. El inconveniente es que hay que explorar sistemáticamente todas las alternativas, lo que resulta muy lento en general.

Como hemos visto, un algoritmo de backtracking tiene ciertos requerimientos:

* Un espacio de búsqueda, esto es un conjunto de posibilidades o "estados" del problema a resolver. En general, un estado puede estar definido por el elemento de la estructura de datos que queremos tratar en cada instante. Por ejemplo, en un árbol, cada uno de sus nodos.
* Un estado inicial, el punto de partida del problema.
* Un conjunto de estados finales. Estos se definen mediante alguna característica, criterio o condición. Por ejemplo, todos los nodos de un árbol que almacenen un número entero igual a 10.
* Una memoria de estados. Generalmente una pila donde guardamos los estados de búsqueda, es decir las alternativas posibles en cada momento.

**ACTIVIDADES**

1. **Ubicación de recursos**
2. El grupo está conformado por Alex Chicaiza y Maria Belen Ceron.
3. Se utilizo DEV C++
4. **Planteamiento del problema**

Realizar cualquier algoritmo que tenga el uso de backtracking de manera fácil y eficiente.

1. **Entregable (s)**

**Clase Arreglo.h**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Module: Arreglo.h

\* Author: Alex Chicaiza, Maria Belen Ceron

\* Modified: Tuesday, November 5, 2019 10:35:18 PM

\* Purpose: Declaration of the class Arreglo

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#if !defined(\_\_Class\_Diagram\_1\_Arreglo\_h)

#define \_\_Class\_Diagram\_1\_Arreglo\_h

class Arreglo

**{**

public**:**

Arreglo**();**

**~**Arreglo**();**

int tamanioCadena**(**char **\*** frase**);**

void invertirCadena**(**char **\*** frase**,** char**\*** fraseInvertida**,** int inicio**,** int fin**);**

protected**:**

private**:**

**};**

#endif

**Clase Arreglo.cpp**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Module: Arreglo.cpp

\* Author: Alex Chicaiza, Maria Belen Ceron

\* Modified: Tuesday, November 5, 2019 10:35:18 PM

\* Purpose: Implementation of the class Arreglo

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include "Arreglo.h"

#include<stdlib.h>

#include <stdio.h>

////////////////////////////////////////////////////////////////////////

// Name: Arreglo::Arreglo()

// Purpose: Implementation of Arreglo::Arreglo()

// Return:

////////////////////////////////////////////////////////////////////////

Arreglo**::**Arreglo**()**

**{**

**}**

////////////////////////////////////////////////////////////////////////

// Name: Arreglo::~Arreglo()

// Purpose: Implementation of Arreglo::~Arreglo()

// Return:

////////////////////////////////////////////////////////////////////////

Arreglo**::~**Arreglo**()**

**{**

// TODO : implement

**}**

////////////////////////////////////////////////////////////////////////

// Name: Arreglo::tamanioCadena(char \* frase)

// Purpose: Implementation of Arreglo::tamanioCadena()

// Parameters:

// - frase

// Return: int

////////////////////////////////////////////////////////////////////////

int Arreglo**::**tamanioCadena**(**char **\*** frase**)**

**{**

**if(\***frase**!=**'\0'**){**

frase**++;**

**return** 1**+**tamanioCadena**(**frase**);**

**}else{**

**return** 0**;**

**}**

**}**

////////////////////////////////////////////////////////////////////////

// Name: Arreglo::invertirCadena(char \* frase, char\* fraseInvertida, int inicio, int fin)

// Purpose: Implementation of Arreglo::invertirCadena()

// Parameters:

// - frase

// - fraseInvertida

// - inicio

// - fin

// Return: void

////////////////////////////////////////////////////////////////////////

void Arreglo**::**invertirCadena**(**char **\*** frase**,** char**\*** fraseInvertida**,** int inicio**,** int fin**)**

**{**

**if(**fin**>-**1**){**

**\*(**fraseInvertida**+**inicio**)** **=**frase**[**fin**];**

inicio**++;**

fin**--;**

invertirCadena**(**frase**,**fraseInvertida**,**inicio**,**fin**);**

**}else{**

**\*(**fraseInvertida**+**inicio**)=**'\0'**;**

**}**

**}**

**Clase PermutacionLetra.h**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Module: PermutacionLetra.h

\* Author: Alex Chicaiza, Maria Belen Ceron

\* Modified: Wednesday, November 6, 2019 3:57:40 PM

\* Purpose: Declaration of the class PermutacionLetra

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#if !defined(\_\_Class\_Diagram\_2\_PermutacionLetra\_h)

#define \_\_Class\_Diagram\_2\_PermutacionLetra\_h

//#include "Arreglo.h"

#include <iostream>

#include <fstream>

#include <string.h>

#include <cstdlib>

#include <cmath>

class PermutacionLetra

**{**

public**:**

std**::**fstream enter**;**

void permutacion**(**std**::**string aux**,**char **\***items**,**int n**,**int r**);**

int tamanioCadena**(**char **\*** frase**);**

protected**:**

private**:**

void generarPermutacion**(**std**::**string aux**,**char **\***items**,**int n**,**int r**);**

**};**

#endif

**Clase PermutacionLetra.cpp**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Module: PermutacionLetra.cpp

\* Author: Alex Chicaiza, Maria Belen Ceron

\* Modified: Wednesday, November 6, 2019 3:57:40 PM

\* Purpose: Implementation of the class PermutacionLetra

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include "PermutacionLetra.h"

#include <iostream>

#include <fstream>

#include <cstdlib>

#include <cmath>

//std::fstream enter;

////////////////////////////////////////////////////////////////////////

// Name: PermutacionLetra::permutacion()

// Purpose: Implementation of PermutacionLetra::permutacion()

// Return: void

////////////////////////////////////////////////////////////////////////

void PermutacionLetra**::**permutacion**(**std**::**string aux**,**char **\***items**,**int n**,**int r**){**

enter**.**open**(**"solucion.txt"**,**std**::**fstream**::**out**);** //para leer in, para salir es out escribir

enter**<<**"\n\n\tUNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE\n"**<<**std**::**endl**;**

enter**<<**"\t INGENIERIA DE SOFTWARE"**<<**std**::**endl**;**

enter**<<**"Ingrantes: "**<<**std**::**endl**;**

enter**<<**"\t\tMaria Belen Ceron"**<<**std**::**endl**;**

enter**<<**"\t\tAlex Chicaiza"**<<**std**::**endl**;**

enter**<<**"\n\t\t\tSOLUCION "**<<**std**::**endl**<<**std**::**endl**;**

enter**<<**"\n\n Numero de permutaciones con repeticion: "**<<**pow**(**n**,**r**)<<**std**::**endl**;**

enter**<<**"\n\n Lista de permutaciones con repeticion:"**<<**std**::**endl**;**

/\*char \*\*arch=(char\*\*)malloc(9\*sizeof(char));

for(int i=0;i<9;i++){

\*arch=(char\*)malloc(sizeof(char));

}\*/

generarPermutacion**(**aux**,**items**,**n**,**r**);**

//enter<<"hola"<<std::endl;

//enter<<"hola"<<std::endl;

//;

enter**.**close**();**

**}**

////////////////////////////////////////////////////////////////////////

// Name: PermutacionLetra::generarPermutacion()

// Purpose: Implementation of PermutacionLetra::generarPermutacion()

// Return: void

////////////////////////////////////////////////////////////////////////

void PermutacionLetra**::**generarPermutacion**(**std**::**string aux**,**char **\***items**,**int n**,**int r**)**

**{**

char**\*** resp**=(**char **\*)**aux**.**c\_str**();**

**if(**r**>**0**){**

**for(**int x**=**0**;** x**<**n**;** x**++){**

generarPermutacion**(**aux**+\*(**items**+**x**),**items**,**n**,**r**-**1**);**

**}**

**}else{**

enter**<<**" "**<<**resp**<<**std**::**endl**;**

printf**(**"\n%s"**,**resp**);**

**}**

**}**

**Clase Main**

/\*\*

Universidad de las Fuerzas Armadas ESPE

Alex Chicaiza, Maria Belen Ceron

NRC: 2967

Fecha creacion: 26/10/2019

Fecha ultima modificacion: 7/11/2019

Docente: Ing. Solis

\*/

#include <stdio.h>

#include <windows.h>

#include <iostream>

#include "PermutacionLetra.h"

#include "Ingresar.h"

#include "Arreglo.h"

**using** **namespace** std**;**

void**\*** moverMarquesina**(**void **\***marquesina**);**

void gotoxy**(**short posicionx**,** short posiciony**);**

int main**(){**

pthread\_t hilo1**;**

pthread\_t hilo2**;**

pthread\_create**(&**hilo1**,NULL,**moverMarquesina**,NULL);**

main**;**

cout**<<**"\n\n\n\n\n\n\n"**;**

DWORD modoConsola**;**

INPUT\_RECORD evento**;**

BOOL SALIR **=** FALSE**;**

Ingreso lectura**;**

Arreglo arr**;**

PermutacionLetra ejecutar**;**

/\*\*

\*Esta funcion sirve para capturar los numeros del teclado

\* \*/

HANDLE hstdin **=** GetStdHandle**(** STD\_INPUT\_HANDLE **);**

/\*\*

\*Sirve para mantener el diseño de la consola

\*\*/

GetConsoleMode**(** hstdin**,** **&**modoConsola **);**

/\*\*

\*Con esta funcion identifica las teclas especiales como f1

\*/

SetConsoleMode**(** hstdin**,** 0 **);**

cout**<<**"\t\tUNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE\n\* Alex Chicaiza\n\* Maria Belen Ceron\nNRC: 2967\nDocente: Ing. Fernando Solis"**<<**endl**;**

//system("cls");

cout**<<**"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"**<<**endl**;**

cout**<<**"\*\tANTES DE USAR LEA LA AYUDA PRESIONANDO f1\t\*"**<<**endl**;**

cout**<<**"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"**<<**endl**;**

cout**<<**"\t\tMENU DE OPCIONES"**<<**endl**;**

cout**<<**"\n1.Generar PDF de permutacion\n2.Comenzar permutaciones"**<<**endl**;**

**while** **(!**SALIR**){**

**if** **(**WaitForSingleObject**(** hstdin**,** 0 **)** **==** WAIT\_OBJECT\_0**){** /\* if kbhit \*/

DWORD count**;**

/\*\*

\*Captura los eventos al presionar una tecla

\*/

ReadConsoleInput**(** hstdin**,** **&**evento**,** 1**,** **&**count **);**

**}**

**if** **((**evento**.**EventType **==** KEY\_EVENT**)** **&&** **!**evento**.**Event**.**KeyEvent**.**bKeyDown**){**

**switch** **(**evento**.**Event**.**KeyEvent**.**wVirtualKeyCode**){**

**case** VK\_ESCAPE**:**

SALIR **=** TRUE**;**

**break;**

**case** VK\_F1**:**

system**(**"Ayuda.hnd"**);**

**break;**

**case** VK\_NUMPAD1**:**

system**(**"txt2pdf.exe solucion.txt respuestas.pdf -oao -pfs60 -pps43 -ptc0 -width3000 -height2000"**);**

**break;**

**case** VK\_NUMPAD2**:**

pthread\_create**(&**hilo1**,NULL,NULL,NULL);**

system**(**"cls"**);**

char**\*** items**=**lectura**.**ingresarLetras**(**"Ingrese las letras a permutar: "**);**

int n**,**r**;**

n**=**arr**.**tamanioCadena**(**items**);**

system**(**"cls"**);**

r**=**lectura**.**ingresarEnteros**(**"Grupo de permutacion: "**);**

system**(**"cls"**);**

cout**<<**"\n Lista de elementos:\n "**;**

**for(**int x**=**0**;** x**<**n**;** x**++){**

cout**<<**items**[**x**]<<**","**;**

**}**

cout**<<**"\n\n Numero de permutaciones con repeticion: "**<<**pow**(**n**,**r**);**

cout**<<**"\n\n Lista de permutaciones con repeticion:\n "**;**

ejecutar**.**permutacion**(**""**,**items**,**n**,**r**);**

Sleep**(**7000**);**

system**(**"cls"**);**

**return** main**();**

**break;**

**}**

evento**.**Event**.**KeyEvent**.**wVirtualKeyCode**=-**1**;**

**}**

**}**

**return** 0**;**

**}**

void gotoxy**(**short posicionx**,** short posiciony**)** **{**

COORD coordenadaPosicion **=** **{**posicionx**,** posiciony**};**

SetConsoleCursorPosition**(**GetStdHandle**(**STD\_OUTPUT\_HANDLE**),** coordenadaPosicion**);**

**}**

void**\*** moverMarquesina**(**void **\***marquesina**){**

int a**,** b**,** c**,** n**=**0**,**letra**=**39**,**pos**=**1**,**cont**=**39**,**aux**,**cont1**=**39**,**auxg**;**

char t**[**50**]** **=**"BIENVENIDO AL PROGRAMA DE PERMUTACIONES\n"**,**auxt**[**39**]=**" "**;**

**do{**

**for** **(**a**=**0**;**a**<**42**;**a**++){**

aux**=**pos**;**

**for(**b**=**39**;**b**>**cont**;**b**--){**

gotoxy**(**pos**,**1**);**

cout**<<**t**[**b**];**

pos**--;**

**}**

aux**++;**

pos**=**aux**;**

cont**--;**

Sleep **(**75**);**

**if(**a**==**40**){**

**break;**

**}**

**}**

**for(**a**=**3**;**a**<**70**;**a**++){**

gotoxy**(**a**-**1**,**1**);**

cout**<<**" "**;**

gotoxy**(**a**,**1**);**

cout**<<**t**;**

Sleep **(**75**);**

**}**

pos**=**70**;**

auxg**=**69**;**

**for** **(**a**=**0**;**a**<**41**;**a**++){**

gotoxy**(**auxg**,**1**);**

cout**<<**" "**;**

aux**=**pos**;**

**for(**b**=**0**;**b**<=**cont1**;**b**++){**

gotoxy**(**pos**,**1**);**

cout**<<**t**[**b**];**

pos**++;**

**}**

cont1**--;**

aux**++;**

pos**=**aux**;**

auxg**++;**

Sleep **(**75**);**

**}**

cont1**=**39**;**

letra**=**39**;**

pos**=**1**;**

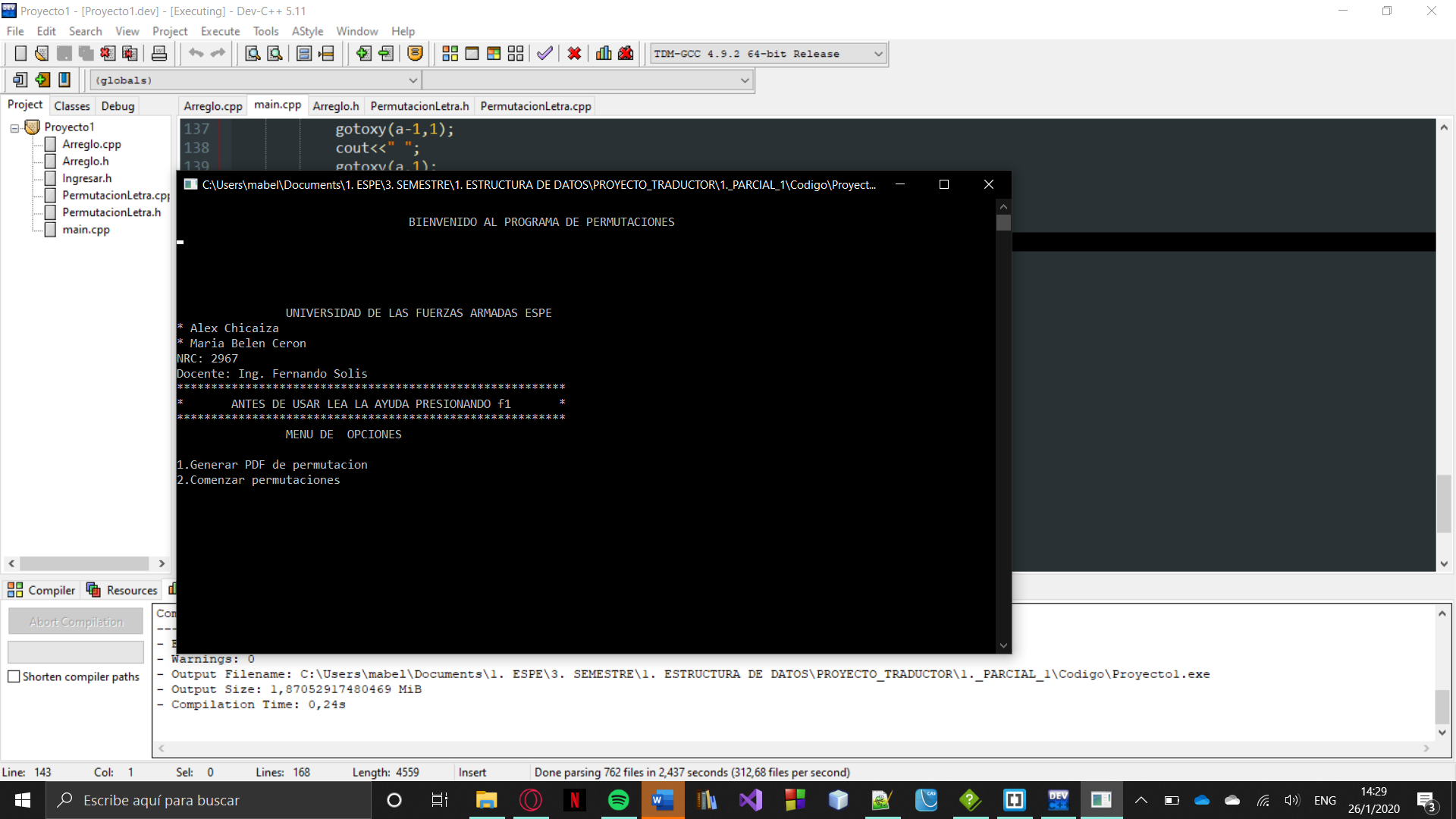
cont**=**39**;**

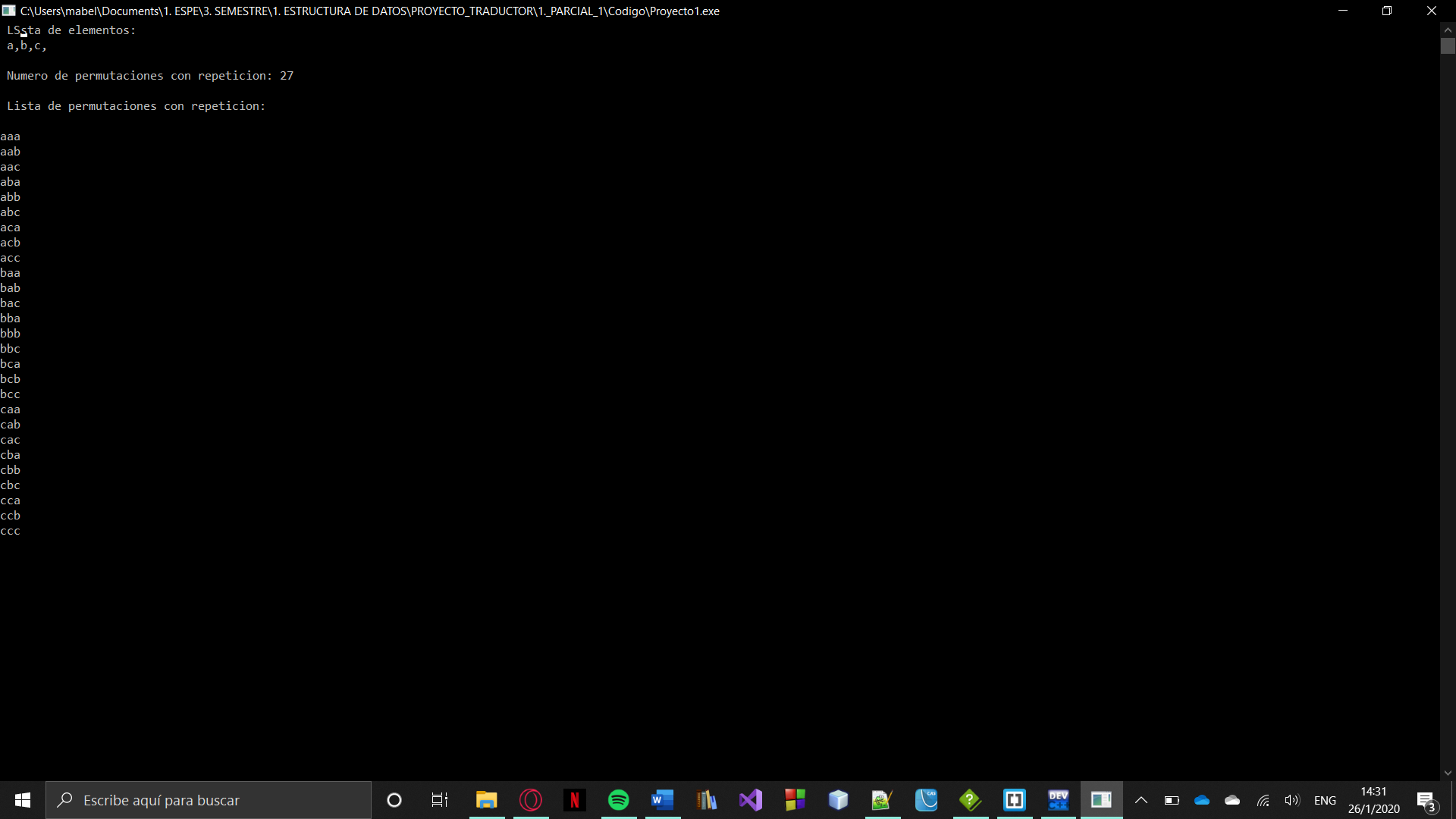
c**=**0**;**

**}while** **(**c**=**1**);**

**}**

1. **Ejecución**





1. **Conclusiones**

Después de las investigaciones y práctica, se pudo manejar de manera más eficiente algoritmos, por ejemplo, de búsqueda, entre otros. Se aprendió a utilizar de manera los algoritmos de backtracking y mejor la recursividad en sí. Se concluye que los algoritmos que usan backtracking son más rápidos que los normales.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

DOCENTE RESPONSABLE COORDINADOR DE ÁREA

Ing. Fernando Solis. MsC. PhD. Rodrigo Fonseca.