

A person with long dark hair, seen from behind, is walking away on a wooden pier at night. The pier has a metal railing on the left and a wooden bench on the right. In the background, there are blurred lights from buildings and street lamps, suggesting a coastal city at night. The overall mood is contemplative and solitary.

El viajero

"Buscando el menor coste"

España



CONTEXTO DEL PROBLEMA

El profesor Franci viajara a España debería recorrer todas las ciudades empezando en Murcia y terminando en la capital Murciana

COSTE MENOR A 22000

ALMACENAR EL RESULTADO
ARCHIVO "OUTPUT.TXT"



Proyecto AED

Implementación

Codificación en C++



Clase Grafo

```
class Grafo
{
    int **ruta;           //para la matriz de rutas
    int **matriz;         //para la matriz de distancias
    Lista<int> ruta2;      //para guardar la ruta de los nodos
    int n_vertices;       //numero de vertices
    int longitud;         //coste del ciclo

    fstream output;       //para el archivo de salida

public:
    Grafo(int);
    void add_arista(int, int, int);
    void floyd_warshall();
    int min_ciclo(int, int, bool);
    int get_ruta(int, int);
    void salida();
};
```

```

Grafo::Grafo(int n)
{
    n_vertices = n;
    //reservar memoria
    matriz = new int *[n_vertices + 1];
    for (int i = 1; i <= n_vertices; i++)
    {
        matriz[i] = new int[n_vertices + 1];
        for (int j = 0; j <= n_vertices; j++)
        {
            matriz[i][j] = 0;
        }
    }

    ruta = new int *[n_vertices + 1];
    for (int i = 1; i <= n_vertices; i++)
    {
        ruta[i] = new int[n_vertices + 1];
        for (int j = 0; j <= n_vertices; j++)
        {
            ruta[i][j] = 0;
        }
    }
}

```

01 Constructor de clase

Las variables miembro **matriz** y **ruta** se les reservara memoria dado que son matrices y se les declara como valores 0..

02 Algoritmo de Warshall

Primera parte

Primer condicional

Si en nuestra matriz de adyacencia esta en la diagonal principal o si el valor de la posición **matriz[i][j]** es **0**. Entonces, el valor de la matriz **matriz[i][j]** toma un valor infinito **INF**.

Segundo condicional

Si en nuestra matriz de adyacencia el valor de **matriz[i][j]** es un valor infinito **INF**. Entonces, el valor de la **ruta[i][j]** tomara un valor infinito **-1** que indicara que la ruta no existe para **ruta[i][j]**.

Tercer condicional

Caso contrario al segundo caso **ruta[i][j]** tomara el valor de **j** "el nodo apuntado por **i**".

```
void Grafo::floyd_warshall()
{
    for (int i = 1; i <= n_vertices; i++)
    {
        for (int j = 1; j <= n_vertices; j++)
        {
            if (i != j && matriz[i][j] == 0)
                matriz[i][j] = INF;

            if (matriz[i][j] == INF)
                ruta[i][j] = -1;
            else
                ruta[i][j] = j;
        }
    }

    for (int k = 1; k <= n_vertices; k++)
    {
        for (int i = 1; i <= n_vertices; i++)
        {
            for (int j = 1; j <= n_vertices; j++)
            {
                if (matriz[i][k] == INF || matriz[k][j] == INF)
                    continue;

                if (matriz[i][j] > matriz[i][k] + matriz[k][j])
                {
                    matriz[i][j] = matriz[i][k] + matriz[k][j];
                    ruta[i][j] = ruta[i][k];
                }
            }
        }
    }
}
```



```
void Grafo::floyd_warshall()
{
    for (int i = 1; i <= n_vertices; i++)
    {
        for (int j = 1; j <= n_vertices; j++)
        {
            if (i != j && matriz[i][j] == 0)
                matriz[i][j] = INF;

            if (matriz[i][j] == INF)
                ruta[i][j] = -1;
            else
                ruta[i][j] = j;
        }
    }

    for (int k = 1; k <= n_vertices; k++)
    {
        for (int i = 1; i <= n_vertices; i++)
        {
            for (int j = 1; j <= n_vertices; j++)
            {
                if (matriz[i][k] == INF || matriz[k][j] == INF)
                    continue;

                if (matriz[i][j] > matriz[i][k] + matriz[k][j])
                {
                    matriz[i][j] = matriz[i][k] + matriz[k][j];
                    ruta[i][j] = ruta[i][k];
                }
            }
        }
    }
}
```

03 Algoritmo de Warshall

Segunda parte

Primer condicional

Si en nuestra matriz de adyacencia **matriz[i][k]** o **matriz[k][j]** tiene un valor **INF**

Entonces, continúe la ejecución del código

Segundo condicional

Si en nuestra matriz de adyacencia el valor de la matriz **matriz[i][j]** es mayor entonces.

Entonces, el valor de la **ruta[i][j]** actualizará el valor con la suma de **ruta[i][k]** y **ruta[k][j]**. Por último **ruta[i][j]** tomará el valor de **ruta[i][k]**.

```

int Grafo::min_ciclo(int begin_node, int second_node, bool
status)
{
    longitud = matriz[begin_node][second_node];
    if (status)
    {
        ruta2.insertar(begin_node);
        ruta2.insertar(second_node);
    }

    bool visited[n_vertices + 1];
    for (int i = 1; i <= n_vertices; i++)
    {
        visited[i] = false;
    }

    int q;
    q = second_node;
    visited[second_node] = true;

    int vis, min = INF;
    int cambio = 1;
    //Para encontrar un arbol de expansion minima
    while (q != 0 && cambio < n_vertices)
    {
        vis = q;
        q = 0;

        for (int i = 1; i <= n_vertices; i++)
        {
            if (matriz[vis][i] != 0 && (!visited[i]))

```

04 Algoritmo de Prim

Primera parte

El atributo de clase **longitud** es lo que retornada en nuestra función.

Si **status** es **true** nuestro agregara a los dos nodos al vector **ruta2** "**vector de nuestro ciclo de menor coste**".

05

Algoritmo de Prim

Segunda parte

Dentro de nuestro for comparamos

- **Primer condicional** :Lo cual indicaría que si ese nodo no ha sido visitado y su valor es diferente de nodo pasa al siguiente condicional.
- **Condicional anidado** : Es menor entonces se actualizan los valores con los valores mínimos.
anidado

```

int cambio = 1,
//Para encontrar un arbol de expansion minima
while (q != 0 && cambio < n_vertices)
{
    vis = q;
    q = 0;

    for (int i = 1; i <= n_vertices; i++)
    {
        if (matriz[vis][i] != 0 && (!visited[i]))
        {
            if (matriz[vis][i] < min)
            {
                q = i;
                min = matriz[vis][i];
            }
        }
    }
    if (status)
        ruta2.insertar(q);
    visited[q] = true;
    cambio++;
    longitud = longitud + matriz[vis][q];
    min = INF;
}
if (status)
    ruta2.insertar(begin_node);
longitud = longitud + matriz[q][begin_node];
return longitud;
}

```

05 Algoritmo de Prim

Tercera parte

Antes de salir del while:

1. Nuestro arreglo booleano **visited** se actualiza.
2. **cambio** incrementa
3. **longitud** se actualiza con el valor de la matriz de adyacencia mínima.
4. Por ultimo **min** toma el valor de infinito para volver encontrar el menor nodo en nuestro for.

Por ultimo, nuestra **longitud** tomara el anterior nodo con nuestro nodo de inicio **712** para volver al inicio de nuestro ciclo. Y obtenemos el coste de mínimo de nuestro ciclo.

```
int vis, min = INF;
int cambio = 1;
//Para encontrar un arbol de expansion minima
while (q != 0 && cambio < n_vertices)
{
    vis = q;
    q = 0;

    for (int i = 1; i <= n_vertices; i++)
    {
        if (matriz[vis][i] != 0 && (!visited[i]))
        {
            if (matriz[vis][i] < min)
            {
                q = i;
                min = matriz[vis][i];
            }
        }
    }
    if (status)
        ruta2.insertar(q);
    visited[q] = true;
    cambio++;
    longitud = longitud + matriz[vis][q];
    min = INF;
}
if (status)
    ruta2.insertar(begin_node);
longitud = longitud + matriz[q][begin_node];
return longitud;
}
```


Resultados y Observaciones



1

Observaciones

1. Como parte del proyecto se nos dio un archivo ***entrada.txt*** según las indicaciones. Se trabajo con el archivo input.txt que es una replica de ***entrada.txt*** sin **N** lineas.
2. El costo computacional de la funcion miembro ***void floyd_warshall*** de la clase Graph es de n^3 , n^2 , el cual para un entorno profesional "***Code Interview***" no es recomendable.

2

Resultado

El costo que se obtuvo en este reto es de 25186.

Fondo de color



Tipografías

Titulos/Helvetica Neue

PARRAFOS/ Helvetica Neue | Estilo de parrafo: Cuerpo Pequeño