PROGRAMA 2– Algoritmo de DIJKSTRA

Asignatura: INTELIGENCIA ARTIFICIAL
Profesor: D. Sc. Gerardo García Gil
2022-B
López Arellano Ricardo David
Universidad de Guadalajara (CUCEI)

Presentación
En este documento hablaremos sobre el algoritmo de Dijkstra el cual te permite calcular la ruta más corta entre un nodo (tú eliges cuál) y todos los demás nodos en el grafo.

Edgar W. Dijkstra (1930 - 2002) nació en Holanda. Fue uno de los primeros en proponer la programación como ciencia. Ganó el prestigioso premio Turing Award de la Association for Computing Machinery en 1972.

Introducción
El algoritmo de dijkstra determina la ruta más corta desde un nodo origen hacia los demás nodos para ello es requerido como entrada un grafo cuyas aristas posean pesos.

Trata de ir explorando todos los caminos más cortos que parten de un vértice origen y recorre todos los demás vértices. Cuando se obtiene el camino más corto desde el vértice origen, al resto de vértices que componen el grafo, el algoritmo se detiene.

Antecedentes
El algoritmo de Dijkstra es un algoritmo eficiente (de complejidad O (n2), donde “n” es el número de vértices) que sirve para encontrar el camino de coste mínimo desde un nodo origen a todos los demás nodos del grafo. Fue diseñado por el holandés Edsger Wybe Dijkstra en 1959.

Desarrollo
1. Seleccionar vértice de partida, es decir un origen.
2. Marcar el punto de partida como el punto de inicio.
3. Determinar los caminos especiales desde el nodo de partida, es decir, el de inicio.
4. Camino especial es aquel que solo puede trazarse a través de los nodos o vértices ya marcados.

5. Para cada nodo no marcado, se debe determinar si es mejor usar el camino especial antes calculado o si es mejor usar el nuevo camino especial que resulte al marcar este nuevo nodo.
6. Para seleccionar un nuevo nodo no marcado como referencia, deberá tomarse aquel cuyo camino especial para llegar a él es el mínimo, por ejemplo si anteriormente marqué el nodo o vértice 2, el cual tiene dos nodos adyacentes 3 y 4 cuyo peso en la arista 4 Programación IV. Guía No. 10 corresponde a 10 y 5 respectivamente, se tomará como nuevo nodo de partida el 4, ya que el peso de la arista o camino es menor.
7. Cada camino mínimo corresponde a la suma de los pesos de las aristas que forman el camino para ir del nodo principal al resto de nodos, pasando únicamente por caminos especiales, es decir nodos marcados.

Implementación

```
//LOPEZ ARELLANO RICARDO DAVID

#include <iostream>
#include <limits.h>
#define max 5

using namespace std;

int matrix[max][max] = {//lo mencionado son los vertices
    // 1, 2, 3, 4, 5
/*1*/{ 0, 100, 30, 0, 0},
/*2*/{ 0, 0, 20, 0, 0},
/*3*/{ 0, 0, 0, 10, 60},
/*4*/{ 0, 15, 0, 0, 50},
/*5*/{ 0, 0, 0, 0, 0}
};

int origen = 0;
int distancia[max];
bool array[max] = { false, false, false, false, false };

int minimaDistancia(){
    int min = INT_MAX;
    int minIndex;
    for(int i = 0; i < max; i++){
        if(array[i] == false && distancia[i] <= min){
            min = distancia[i];
            minIndex = i;
        }
    }
    return minIndex;
}

int mostrar (){
        cout << "\n\t < ALGORITMO DE DIJKSTRA >"<<endl<<endl;
    cout << "Distancia: " << endl;
```

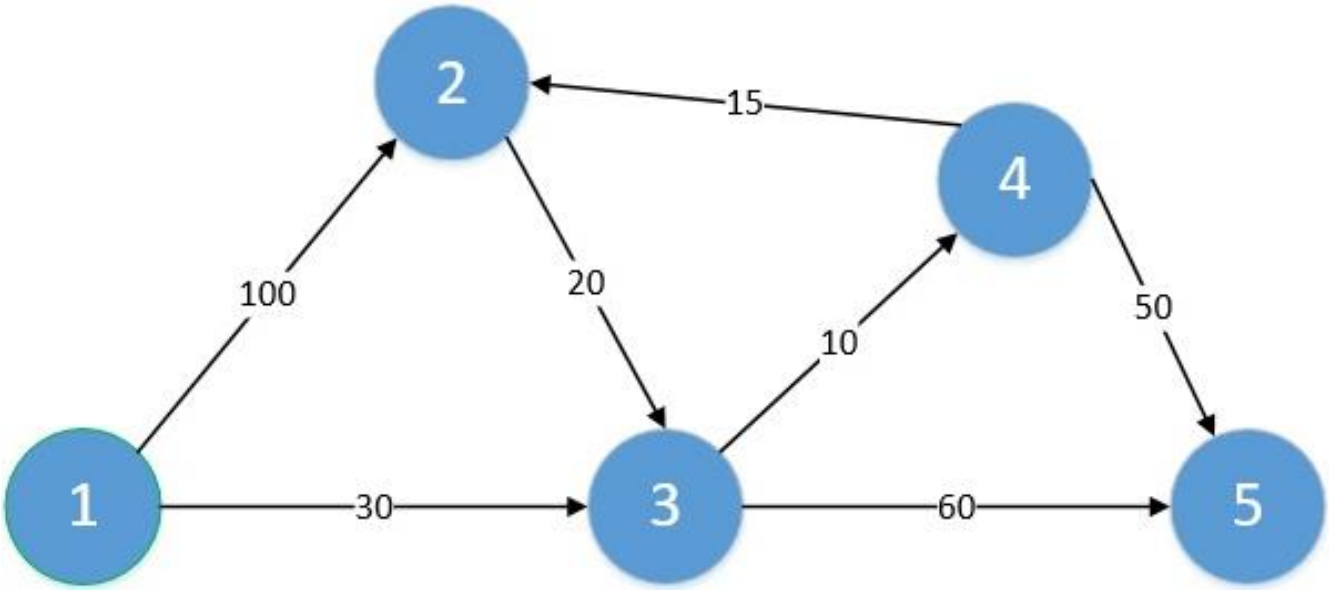
```
    for(int i = 0; i < max; i++){
        cout << "Nodo:" << i << " Distancia: " << distancia[i] << endl;
    }
}

void dijkstra(){
    for(int i = 0; i < max; i++){
        distancia[i] = INT_MAX;
        distancia[origen] = 0;
    }

    for(int i = 0; i < max-1; i++){
        int u = minimaDistancia();
        array[u] = true;

        for(int j = 0; j < max; j++){
            if(!array[j] && matrix[u][j] && distancia[u] != INT_MAX && distancia[u] + matrix[u][j] <
distancia[j]){
                distancia[j] = distancia[u] + matrix[u][j];
            }
        }
    }

    mostrar();

}

int main(int argc, char const *argv[])
{
    dijkstra();

    return 0;
}
```

Experimentos

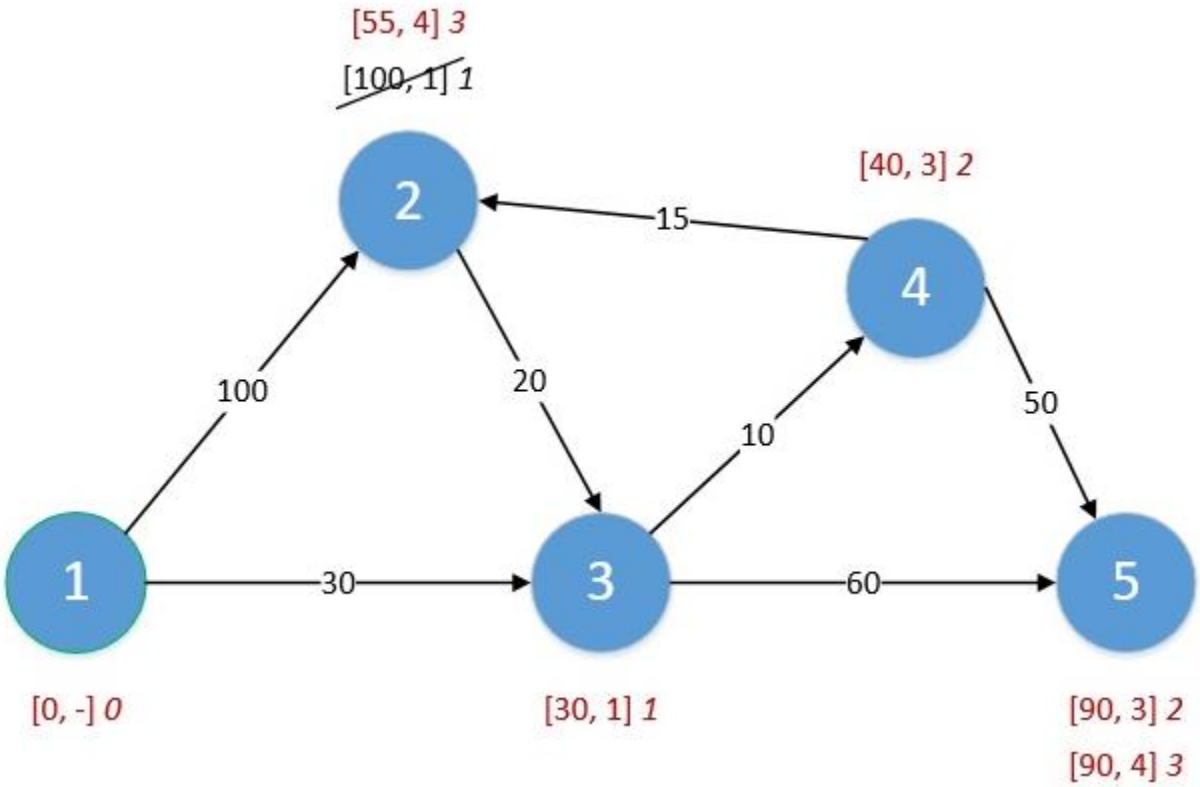
Para lograr programar el algoritmo de Dijkstra lo realicé en C++ ya que siento que ahí soy más fuerte y podía realizarlo de una mejor manera.

Resultados

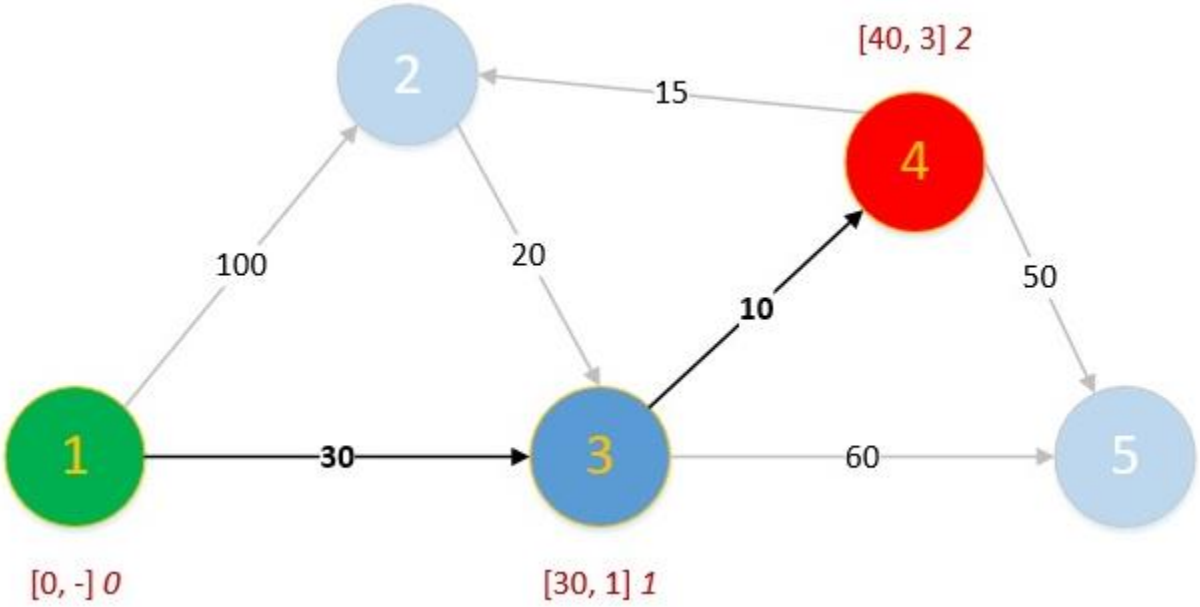
Utilicé la siguiente red como ejemplo:



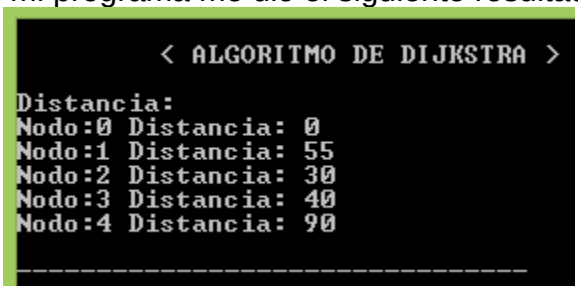
Haciendo el ejercicio de Dijkstra da el siguiente resultado en cada nodo:



El siguiente resultado es de la ruta más corta:



En mi programa me dio el siguiente resultado:



Conclusión

El algoritmo de Dijkstra siempre tiene una solución optima a este tipo de problemas donde se busca el árbol de expansión mínimo eso quiere decir que este algoritmo pertenece a P porque se puede resolver de forma eficiente por una maquina determinista en tiempo polinomial.

Tuve algo de complicaciones para programarlo para a fin de cuentas se pudo sacar el programa funcionando correctamente.

Referencias

1. Micha, E. (2003). Matemáticas Discretas.México: Editorial LIMUSA, S.A.
2. Calderón, H.D (2008). Matemáticas Discretas para la cinecia de Comunicación. Puno,Perú: Editorial Pacifico.
3. Espinosa, R (2010). Matemáticas Discretas Mexico: Alfaomega Grupo Editor