



Instituto Politécnico Nacional
Unidad Profesional Interdisciplinaria en
Ingeniería y Tecnologías Avanzadas



Ingeniería en Telemática
Transmisión de Datos.

Practica 3. Ruteo

ALUMNO

González Serrano Luis Felipe

PROFESORA

Villordo Jiménez Iclia.

Fecha de entrega

26/01/2021

PSEUDOCODIGO

1. Inicializamos el grafo
2. Cada Nodo tendra las poinderaciones mostradas en el diagrama proporcionado por la profesora
3. Se inicializa una matriz que contendra los valores obtenidos del algoritmo de disjktra
4. Establecer un nodo raiz o nodo inicial
5. Establecemos la distancia del nodo actual como 0 para cada NODO y para los nodos que no intersecctan con el nodo actual como infinito
6. Recorremos el grafo de un inicia
 - a. Tomamos el valor inicial del nodo actual
 - b. Comprueba los nodos vecinos
 - c. Sumamos las ponderaciones con el primer nodo vecino
 - d. Se compara con el nodo actual y se elige la minima
 - e. Se marca el nodo como recorrido
 - f. Pasa al siguiente nodo
 - g. Si no hay mas nodos vecinos entonces volvemos
7. Se establece al siguiente nodo como el nodo actual
8. Se almacena el nodo anterior como nodo recorrido e
9. Se retgistra en una tabla
10. Se repiten el paso 6 al 9
11. Si ya no hay nodos que recorrer
12. Termina el proceso

De manera breve y de forma enlistada se explico el proceso con el cual se realiza el proceso de disjktra y como se almancenan los diferentes valores que obtenemos al hacer el recorrido obteniendo la distancia minima en cada nodo. Para obtener el grafo se utilizo una matriz en la cual denominamos a cada nodo como el nombre de la estacion, en este caso cada estacion o nodo esta nombrado por una letra por facilidad y este vector contiene las ponderaciones de todos los nodos, tanto de los nodos vecinos, los cuales son los valores ponderados que conectan a dicho nodo y a los que no estan coenctados directamente a estos se les denomino como infinito su distancia, para facilitar la comparacion y asi obtener la distancia minima mas facilmente y por supuesto el nodo actual se le denomino como cero a todos estos se les ingreso a una matriz y asi de esta forma simulamos el grafo.

DESARROLLO

Del siguiente grafo que simula la conexión de diversas estaciones ponderada con diferentes distancias. Se desea conocer la distancia mínima que puede realizar cada estación al transmitir datos

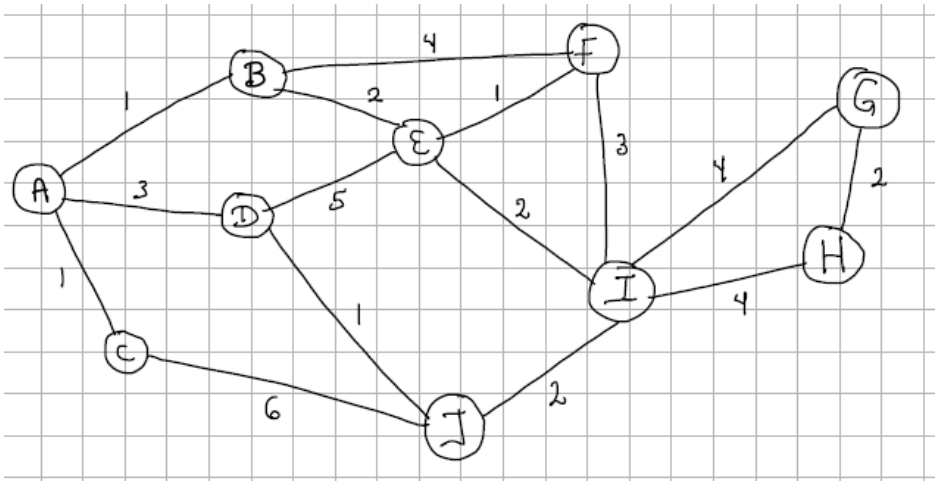


Figura 1. Grafo

Utilizando el algoritmo de Dijkstra pudimos encontrar las rutas mas cortas para la transmisión de datos de cada estación (nodo)

RESULTADOS

```
ruta de 10 a 1
Distancia mínima: 4
```

Los resultados los obtenemos de este tipo junto con una tabla como se muestra en la siguiente imagen

10	1	1041
10	2	10412
10	3	10413
10	4	104
10	5	1095
10	6	1096
10	7	1097
10	8	1098
10	9	109
10	10	10

Interpretando los resultados obtenemos las siguientes tablas

Distancia mínima nodo A

Nodo	Distancia mínima	Ruta
------	------------------	------

A	0	-
B	1	-
C	1	-
D	3	-
E	3	B
F	5	B
G	9	B,E,I
H	9	B,E,I
I	5	B,E
J	4	B,E

Distancia mínima nodo B

Nodo	Distancia mínima	Ruta
A	1	-
B	2	A
C	0	-
D	4	A
E	4	A,B
F	5	A,B
G	8	E,I
H	8	E,I
I	4	E
J	6	E,I

Distancia mínima nodo C

Nodo	Distancia mínima	Ruta
A	1	-
B	1	-
C	1	-
D	3	-
E	3	B
F	5	B
G	10	A,B,E,I
H	10	A, B,E,I
I	6	A,B,E
J	5	A,D

Distancia mínima nodo D

Nodo	Distancia mínima	Ruta
A	1	-
B	1	-
C	1	-
D	3	-
E	3	B
F	5	B
G	10	A,B,E,I
H	10	A, B,E,I

I	6	A,B,E
J	5	A,D

Distancia mínima nodo E

Nodo	Distancia mínima	Ruta
A	3	B
B	2	-
C	4	B,A
D	5	-
E	0	-
F	1	-
G	6	I
H	6	I
I	2	-
J	4	I

Distancia mínima del nodo F

Nodo	Distancia mínima	Ruta
A	4	E,B,A
B	3	E
C	5	E, B,A
D	6	E
E	1	-
F	0	-
G	7	I
H	7	I
I	3	-
J	5	I

Distancia mínima nodo G

Nodo	Distancia mínima	Ruta
A	9	I,E,B
B	8	I,E
C	10	I,E, B,A
D	7	I,J
E	6	I
F	7	I
G	0	-
H	2	-
I	4	-
J	6	I

Distancia mínima nodo H

Nodo	Distancia mínima	Ruta
A	9	I,E,B
B	8	I,E

C	10	I, E, B, A
D	7	I, J
E	6	I
F	7	I
G	2	-
H	0	-
I	4	-
J	6	I

Distancia mínima nodo I

Nodo	Distancia mínima	Ruta
A	5	E, B
B	4	E
C	6	E, B, A
D	3	J
E	2	-
F	3	-
G	4	-
H	4	-
I	0	-
J	2	-

Distancia mínima nodo J

Nodo	Distancia mínima	Ruta
A	4	D
B	5	A
C	5	A
D	1	-
E	5	D
F	5	I
G	6	-
H	6	-
I	2	-
J	0	-

CONCLUSIONES

Como podemos observar el uso de estos algoritmos que facilitan la transmisión de datos son realmente útiles en sistemas prácticos en caso de que se requiera transmitir un mensaje en el menos tiempo posible, si bien ahora es una simulación y simplemente estamos obteniendo las tablas de ruteo con un ejemplo sencillo es posible notar la facilidad de poder interpretar los datos obtenidos que son las tablas y conocer la ruta que debe de tomar cada nodo. En estructura de datos conocí este algoritmo y si bien realice una aplicación en su tiempo y verlo de nuevo ahora en esta materia me da a entender que es un algoritmo utilizado o al menos conocido y útil

CODIGO

```
clc
clear all
close all
%Declaracion del grafo
%%[A B C D E F G H I J]
A=[0 1 1 3 inf inf inf inf inf inf];
B=[1 0 inf inf 2 4 inf inf inf inf];
C=[1 inf 0 inf inf inf inf inf inf 6];
D=[3 Inf Inf 0 5 Inf Inf Inf Inf 1];
E=[inf 2 Inf 5 0 1 Inf Inf 2 Inf];
F=[Inf 4 Inf Inf 1 0 Inf Inf 3 Inf];
G=[Inf Inf Inf Inf Inf Inf 0 2 4 Inf];
H=[Inf Inf Inf Inf Inf Inf 2 0 4 Inf];
I=[Inf Inf Inf Inf 2 3 4 4 0 2];
J=[Inf Inf 6 1 Inf Inf Inf Inf 2 0];
Grafo=[A;B;C;D;E;F;G;H;I;J];
%Recorrido
for i=1:length(Grafo)
    for NodoI=1:length(Grafo)
        for NodoF=1:length(Grafo)
            Rutaminima=Dijkstra(Grafo,NodoI,NodoF);
            TablaRuteo(NodoF,1)=string(NodoI);
            TablaRuteo(NodoF,2)=string(NodoF);
            A=string(horzcat(Rutaminima{:}));
            str=' ';
            for j=1:length(A)
                str=strcat(str,A(j));
            end
            TablaRuteo(NodoF,3)=str;
        end
    end
end

end
function Rutaminima = Dijkstra(Grafo, Inicio, Fin)
%Inicializacion
% Se asigna memoria.
Rutaminima = cell(numel(Inicio), numel(Fin));
distanciaMinima = zeros(numel(Inicio), numel(Fin));

%Matrices con ceros e in
saltos = min(size(Grafo));
PorVisitar = zeros(saltos, 1);
visitados = inf(saltos, 1);

for i = 1:numel(Inicio)

    % Inicializamos las matrices.
    posicion = Inicio(i);
    if issparse(Grafo) % retorna valor logico 1 (true) si la clase es
escasa y logico 0 (false) otro caso.
        index = transpose(full(Grafo(posicion, :) > 0.0));
        index(posicion) = true;
    else
        index = transpose(not(isinf(Grafo(posicion, :)))));
```

```

end
visitados(:) = inf;
visitados(index) = transpose(Grafo(posicion, index));
PorVisitar(:) = 0;
PorVisitar(index) = posicion;
fila = find(index);
while not isempty(fila)
    nodoInicio = fila(1);
    distance = transpose(visitados(nodoInicio) + Grafo(nodoInicio,
:));
    index = distance < visitados;
    if issparse(Grafo)
        index = and(index, full(transpose(Grafo(nodoInicio, :) >
0)));
    end
    if any(index) %si es distinto de cero
        visitados(index) = distance(index);
        PorVisitar(index) = nodoInicio;
        fila = cat(1, fila(2:end, :), find(index));
    else
        fila = fila(2:end, :);
    end
end
for j = 1:numel(Fin)
    Rutaminima{i, j} = zeros(size(PorVisitar));
    nodoFin = Fin(j);
    for nodoInicio = numel(PorVisitar):-1:1
        if or(eq(posicion, nodoFin), eq(nodoFin, 0))
            break
        end
        Rutaminima{i, j}(nodoInicio) = nodoFin;
        nodoFin = PorVisitar(nodoFin);
    end
    if eq(posicion, nodoFin)
        Rutaminima{i, j}(nodoInicio) = nodoFin;
        Rutaminima{i, j} = Rutaminima{i, j}(nodoInicio:end);
    else
        Rutaminima{i, j} = [];
    end
end
distanciaMinima(i, :) = visitados(Fin);
disp("Distancia mínima: "+distanciaMinima);

end
end

```