

Laboratorio Nro. 4

Escribir el tema del laboratorio

Simón Correa Henao
Universidad Eafit
Medellín, Colombia
scorreah@eafit.edu.co

David Gómez Correa
Universidad Eafit
Medellín, Colombia
dgomezc10@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1 La estructura de datos que se utilizó para resolver este problema fue una matriz, en la cual se almacenaba el mapa, es decir todos los nodos y todos los vértices con los que se unían con sus respectivos pesos. Por otro lado, el algoritmo voraz que se utilizó fue el del vecino más cercano, en el cual partiendo del origen se iba preguntado por el vecino más cercano cada vez que se visitaba un nuevo nodo, para el control de este algoritmo se utilizó un array encargado de guardar los nodos visitados y un stack para ir almacenando el nodo actual en el que nos encontrábamos.

3.2 Este algoritmo es muy probable que no retorne la solución óptima, puesto que este no evalúa otras posibles soluciones, es decir no considera más soluciones para así escoger la más adecuada, sino que simplemente escoge el nodo teniendo como criterio el que más cerca se encuentre. El criterio que debe cumplir el grafo para que este arroje una solución es que el grafo sea completo, es decir que entre todos los nodos exista una arista, esto puesto que puede llegar un punto en el que no se visite un nodo puesto que este no está conectado con el nodo actual, o llegar a un nodo sin salida, es decir un nodo que no tiene más de un vecino.

3.3 Para adaptarlo a domicilios en la ciudad de Medellín sería solo considerar los puntos donde hay que entregar domicilios, así de esta manera el tamaño del problema reduciría de manera considerable. Para calcular la distancia de un punto a otro se podría realizar distancia entre dos puntos, trazar una línea recta entre los nodos, y el camino a recorrer sería por las calles más cercanas a esta línea, así de esta manera se podría asegurar que se está recorriendo el camino más corto.

ESTRUCTURA DE DATOS 2
Código ST0247

4) Simulacro de Parcial

4.1 $i = j$

4.2 $min > adjacencyMatrix[element][i];$

4.3 .1

Paso	A	B	C	D	E	F	G	H
1	A	20,A	∞	80,A	∞	∞	90,A	∞
2	B	20,A	∞	80,A	∞	30,B	90,A	∞
3	F	20,A	40,F	70,F	∞	30,B	90,A	∞
4	C	20,A	40,F	50,C	∞	30,B	90,A	60,C
5	D	20,A	40,F	50,C	∞	30,B	70,D	60,C
6	H	20,A	40,F	50,C	∞	30,B	70,D	60,C
7	G	20,A	40,F	50,C	∞	30,B	70,D	60,C
8	E	20,A	40,F	50,C	∞	30,B	70,D	60,C

4.3.2 Camino A-> G: A,B,F,C,D,G (Valor 70)

4.4 .1 $temp / 2$

.2 $minimo + temp$

.3 $O(1)$

4.5 -

4.6 .1 $i + 1$

.2 $res + 1$

.3 $last = i$

.4 Respuesta: 2

PhD. Mauricio Toro Bermúdez

Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473