

Laboratorio Nro. 4

Algoritmos voraces

Juan Pablo Rincon Usma
Universidad Eafit
Medellín, Colombia
jprinconu@eafit.edu.co

Julian Gomez Benitez
Universidad Eafit
Medellín, Colombia
jgomezb11@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1 Usamos listas de adyacencia para representar el grafo que contiene los nodos de nuestro mapa, luego usando el método Greedy llamado el vecino más cercano nos desplazamos desde el nodo de origen hacia los sucesores siguiendo la regla del que esté más cerca, es decir, de todos los sucesores se elige cual es el que tiene menos costo de desplazamiento. Los que se visitan se guardan en una lista para no ser visitados de nuevo.

3.2 No siempre se entrega la solución más optima se entrega la primera que se encuentre. Para que el algoritmo voraz funcione correctamente se necesita que todos los nodos estén conectados entres si de lo contrario podría encontrarse en un callejón sin salida en donde no queden nodos por visitar y no esté actualmente en el nodo donde inicio.

3.3 Pues la idea sería que a partir del negocio (nodo 0) se empiecen a visitar los solo los nodos que solicitaron domicilio usando el mismo criterio de que se escoge el que esté más cerca. En términos técnicos sería agregar una condición dentro de la estructura del algoritmo que valide si el nodo a visitar solicito domicilio, de ser verdadera se procede con el algoritmo voraz de manera normal, de lo contrario se saltaría a la siguiente iteración. Si suponemos que el mapa está representado con un plano cartesiano la distancia sería igual a la distancia manhattan entre dos puntos.

3.4 Usamos arreglos para representar a los conductores y las duraciones de las rutas, debido solo nos concentraremos en acceder a los datos y cambiarlos, no necesitamos buscar datos como tal, por ello decidimos usar arreglos que tiene una complejidad para insertar y acceder de $O(1)$, el algoritmo busca las horas más cortas con el fin de asignársela a un conductor, esta se asigna y automáticamente se pone visitada esa ruta debido a que no se pueden repetir, luego se suman cada una de las rutas almacenadas para cada uno de los conductores y luego se le resta la cantidad d , de la duración de la ruta y si el valor es mayor que 0 es porque el conductor trabajo horas extra, para ello se multiplican por la tarifa r , después se retorna el valor de esta suma.

ESTRUCTURA DE DATOS 2
Código ST0247

3.5 La complejidad de este método es de $O(n+m)$.

3.6 Donde n representa las rutas de la mañana y la m representa las rutas de la tarde.

4) Simulacro de Parcial

4.1 $i = j$;

4.2 $\min > \text{adjacencyMatrix}[\text{element}][i]$;

4.3 No se hizo

4.4 $\text{Temp} = \text{temp}/2$;
 $\text{return } \min + \text{temp}$;
 Complejidad $b = O(1)$

4.5 No se hizo

4.6 $x[i] == \text{last}$
 $\text{res} = \text{res} + 1$
 $\text{last} = i$;
 $\text{salida} = 3$

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

