Proyecto Discretas 2

Entendiendo el problema:

El problema plantea, ya en términos matemáticos, dado un grafo pesado G = (E,V), donde se sabe cuáles vértices pertenecen al conjunto de Bancos, y cuáles vértices pertenecen al conjunto de Estaciones de Policia. Una vez se tiene esto las preguntas a responder son las siguientes: ¿Cuántos, bancos están más alejados de las estaciones de policía? ¿A qué distancia se encuentran? Y ¿Cuáles son dichos bancos?

Estrategia de Solución:

Una vez entendido el problema, se pasa a plantear posibles caminos a tomar, para poder llegar a la solución, como lo que se está pidiendo es dar aquellos bancos cuya distancia mínima sea la máxima, entre todos los bancos, un primer acercamiento es el algoritmo de Dijkstra.

Qué es el algoritmo de Dijkstra:

El algoritmo de Dijkstra es un algoritmo que permite hallar los caminos mínimos desde una fuente a todos los demás vértices de un grafo pesado G = (V,E). es decir, este algortitmo recibiendo un grafo pesado G= (V,E) |V| = n y un v0 talque v0 pertenece a v, retorna una lista dist[0…n) donde for all i (0<= i <n) | dist[i] es la distancia minima en el grafo desde v0 hasta vi.

Cómo funciona el algoritmo de Dijkstra:

```
DIJKSTRA(G, w, s)
1  INITIALIZE-SINGLE-SOURCE(G, s)
2  S = ∅
3  Q = G.V
4  while Q ≠ ∅
5      u = EXTRACT-MIN(Q)
6      S = S ∪ {u}
7      for each vertex v ∈ G.Adj[u]
8          RELAX(u, v, w)
```

Inicializar: se debe mantener registro de las distancias, sin embargo, antes de empezar con el algoritmo no se conocen las distancias desde la fuente a todos los demás vértices, por lo que, se dice que antes de empezar la distancia de la fuente a todos los demás vértices es INFINITO, además, la distancia de la fuente, a la fuente misma es 0.

Iteraciones: Se tiene inicialmente un conjunto S, el cual empieza vacio, en cada iteración lo que se va a hacer es ir añadiendo vértices a S, hasta que este ya contenga todos los vértices de G, (en este pseudocodigo Q = V-S) el procedimiento para añadir un vértice a S es el siguiente:

Se escoge un vértice u que pertenezca a V-S, y que además su distancia sea desde la fuente hasta u, sea mínima, se añade u al conjunto S, y para cada vértice adyacente a u, se va a relajar.

Relajación: relajar un vértice v es cual es adyacente a u, consiste ne lo siguiente:

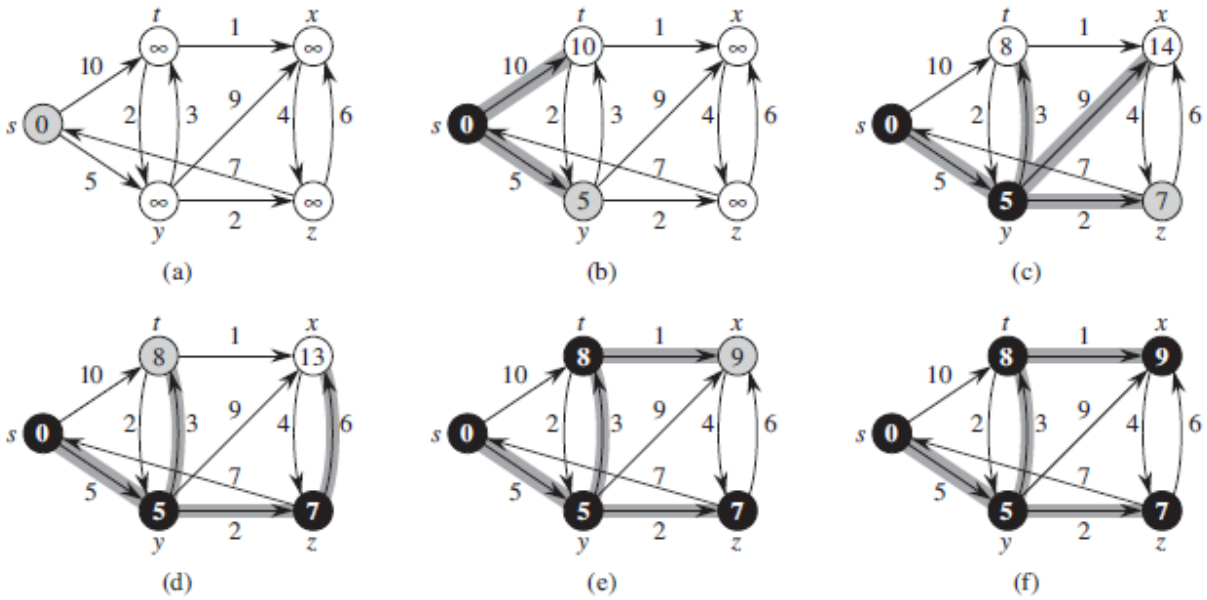
Si la distancia que se sabe hasta ahora que hay desde la fuente hasta u es mayor que la distancia que hay llegando a v a través de u, entonces la distancia desde la fuente hasta v, para a ser la distancia de la fuente hasta u + la distancia de u a v.

En términos más explicitos sea (uv,w) Perteneciente a E, donde v y u pertenecen a V y w perteneciente a los reales positivos, (uv,w) representa que existe un la arista v-u y esta tiene un peso w.

Sea d un registro de distancias desde la fuente hasta todos los vértices de G, d(v) con v perteneciente a V, es la distancia desde la fuente hasta v.

Relajar consiste en: sean u y v vértices de G, y existe la arista (uv,w)

Si d(u) > d(v) + w entonces d(u) pasa a ser d(v) + w



En esta imagen se ve claramente el proceso de relación de los vértices, donde si bien en la figura (b) el vértice t tiene una distancia 10 (d(t) = 10) en la figura (c), esta pasa a ser 8 pues existe la arista y-t y la d(y) + 3 = 8, el cual el menor que 10, por lo que d(t) pasa a ser 8

Como aplicar el algoritmo de Dijkstra al problema Bank Robbery:

Como se dijo anteriormente el algoritmo de Dijkstra se conoce como single-source shortest path, es decir de una sola fuente a todos los demás vértices, sin embargo, el problema Bank Robbery plante múltiples Estaciones de Policía.

Supongamos que se tiene un grafo G = (E,V) con las características que se especifican el problema. Y que de igual forma existe un camino desde la estación p0 a banco b0, donde p0 y b0 pertenecen a V, y un camino de desde la estación p1 al banco b0 donde p1 pertenece a V.

Se vería asi:

p0-…-b0-…-p1

necesariamente se va a llegar antes por un camino que por el otro (o primero se llega viniendo desde p0 o viniendo desde p1), supongamos que se llegó primero por p0, por lo que ya existe un valor en d(b0), ahora, al llegar por p1 hay 2 opciones:

o b0 ya pertenece a S o no pertenece, si ya pertenece al hacer la relajación no se cambiara d(b0), si no pertenece y el camino por p1 es menor d(b0) cambia, si el camino por p1 es mayor d(b0) no cambia.

Al final el valor que quede en d(b0) es el que importa, pues el problema pide el camino más corto desde las estaciones de policía, no el camino más corto desde cada estación a los bancos, es decir si hay 2 caminos a un mismo banco, el que importa es el más corto.

Esto en el algortimo se logra haciendo que en la etapa de inicialización, todos los vértices que estén marcados como Estaciones de Policias, su distancia sea 0, es decir, for all v where v es estación de Policia d(v) = 0.

Implementacion:

Se adjunta un archivo robbery.py en el cual está la implementación del código que resuelve el problema

Conclusiones:

El algortimo de Dijkstra es muy importante en el día a día, muchos enrutadores (routers) utlizan este algoritmo para poder resolver las rutas mas cortas para enviar los paquetes a través de la red. Este proyecto demuestra lo importante que es ser capaz de abstraer la esencia de un algoritmo y poder utilizarlo ese conocimiento para adaptarlo a distintas clases de problemáticas que se puedan presentar.

Juan Manuel Cuellar Borrero

BILBIOGRAFIA:

Introduction to Algorithms 3rd edition - Cormen. T, Leiserson.C, Rivest.R, Stein.C