

OBSERVACIONES DEL LA PRACTICA

David Samuel Rojas Sánchez - Cod 202214621 - ds.rojass1@uniandes.edu.co

Marco Alejandro Ramírez Camacho – Cod 202210308 – ma.ramirez23@uniandes.edu.co

a) ¿Existe alguna diferencia entre los resultados encontrados por BFS y DFS?

Sí, el camino dado por DFS es de 59 nodos, mientras que el BFS tiene 35, siendo sustancialmente más corto. Pero esto tiene un costo, pues el BFS tardó 318.96 ms, siendo mayor a los 291.17 del DFS.

b) ¿Por qué existen diferencias entre los dos algoritmos?

Porque el DFS lo que hace es buscar profundamente en los hijos de una rama específica del grafo, y si no encuentra la respuesta en esta, sigue con la próxima, y al primer camino que encuentre que sirve, lo devuelve. Por otro lado, el BFS consiste en ir mirando por niveles del grafo hasta encontrar la solución (esto lo hace gracias a una queue), y al hacerlo de forma uniforme llegará a la respuesta más optima, pero esto tiene el costo que tiene que hacer todos los caminos a la vez, consumiendo un poco más de tiempo, pero dando la mejor respuesta posible.

i) ¿Cuántos grafos se necesitan definir para solucionar los requerimientos del reto? y ¿Por qué?

Mínimo, es necesario disponer de dos grafos para solucionar los requerimientos del reto. De estos dos grafos, uno debe ser dirigido y el otro no dirigido con pesos. Esto, dado que algunos algoritmos sólo son posibles de ejecutar en grafos no dirigidos (tales como el algoritmo de Prim y Kruskal los cuales calculan los árboles de recubrimiento mínimo o MSTs). Por otra parte, un grafo dirigido que representa bien las rutas de buses de Barcelona ya que las rutas tienen una dirección específica que será útil tener en cuenta al momento de resolver requerimientos que necesiten de algoritmos para grafos dirigidos tales como Dijkstra, el cual encuentra caminos de costo mínimo.

ii) ¿Cuáles son las características específicas de cada uno de los grafos definidos? (vértices, arcos, denso o disperso, dirigido o no dirigido).

Respecto al grafo dirigido, este tendrá como vértices los identificadores para cada estación, es decir, de la forma Code-IDRuta. Con los arcos se representará la secuencia que realizan las rutas de buses, es decir, la conexión que realiza una ruta de bus entre una estación y otra. Además, estos arcos tendrán un peso equivalente a la distancia Harvesine en kilómetros entre las dos estaciones. Por otra parte, se considera que este grafo sería uno disperso, pues el número de arcos que posee está más cercano al de un grafo vacío que al de uno lleno. Esto sucede por la manera en la cual se propone la construcción del grafo dado que, si un bus solo puede realizar una parada a la vez, es decir, los

vértices de nuestro grafo (a excepción de las estaciones de transbordo) tiene como mucho dos arcos que los conectan con la estación previa visitada y la estación destino.

Así mismo, el grafo no dirigido tiene los mismos vértices, arcos y densidad. Con la diferencia de que este posibilita la ejecución de algoritmos propios solamente de los grafos no dirigidos.

iii) Además de los grafos, ¿Qué otras estructuras de datos adicionales se necesitan para resolver los requerimientos? Y ¿Por qué?

Otras estructuras de datos, que serán útiles además de los grafos, podrían ser las tablas de hash y las listas. Con estas estructuras podríamos complementar el grafo y organizar la información que no es posible modelar solamente con este, como es el caso del barrio, el distrito, la longitud y la latitud. Para este tipo de información podría ser eficiente utilizar tablas de hash cuyas llaves tengan el ID de las estaciones, y su valor correspondiente sea la información perteneciente a tal estación. Respecto a las listas, podrían implementarse dos. Una que almacene los vértices y otra que obtenga los arcos. La lista que contiene los arcos es de gran importancia pues con esta es posible extraer los datos únicos para posteriormente calcular la distancia harvesiana. De esta manera uno de los requerimientos podría estar parcialmente desarrollado desde la carga de datos.