

OBSERVACIONES DEL LA PRACTICA

1. Adriana Sofia Roza Cepeda Cod. 202211498
2. Diego Fernando Galván Cruz Cod. 202213709
3. Juan David Castillo Quiroga Cod. 202210669.

Preguntas de análisis

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

En cuanto al tiempo de ejecución BFS y DFS tienen un tiempo de ejecución similar. Cuando hablamos de espacio DFS es eficiente mientras que BFS no lo es. Esto es porque DFS toma espacio lineal porque se tiene una sola ruta con nodos no explorados, mientras que BFS mantiene todos los nodos en la memoria.

Los resultados fueron “satisfactorios”, es decir: En la opción 9 encontré caminos y en la 10 mostré lo que tuvieron que recorrer para llegar al destino. Como era de esperarse el camino de búsqueda del dfs fue mayor con respecto al bfs por las razones que se expondrán en la siguiente pregunta.

Resultados con dfs:

```
>8
Selecione 'dfs' o 'bfs' como algoritmo: dfs
Estación Base: BusStopCode-ServiceNo (Ej: 75009-10): 75009-10
Calculando caminos de busqueda con dfs
FIN

*****
Bienvenido
1- Inicializar Analizador
2- Cargar información de buses de singapur
3- Calcular componentes conectados
4- Establecer estación base:
5- Hay camino entre estacion base y estación:
6- Ruta de costo mínimo desde la estación base y estación:
7- Estación que sirve a mas rutas:
8- Cambiar metodo de busqueda ('bfs' o 'dfs'):
9- Existe un camino de busqueda entre base y estación:
10- Ruta de busqueda entre la estación base y estación:
0- Salir
*****
Selecione una opción para continuar
>9
Estación destino (Ej: 15151-10): 15151-10
Hay camino de busqueda entre la estación base : y la estación: 15151-10:
true
```

```
Selecione una opción para continuar
>10
Estación destino (Ej: 15151-10): 15151-10
El camino de busqueda es de longitud: 60
75009-10
76059-10
76069-10
96289-10
96109-10
85079-10
85089-10
85069-10
85059-10
85049-10
85039-10
85029-10
85019-10
94079-10
94069-10
94059-10
94049-10
94089-10
94039-10
94029-10
94019-10
93099-10
93089-10
```

Resultados con bfs:

```
>8
Seleccione 'dfs' o 'bfs' como algoritmo: bfs
Estación Base: BusStopCode-ServiceNo (Ej: 75009-10): 75009-10
Calculando caminos de busqueda con bfs
FIN

*****
Bienvenido
1- Inicializar Analizador
2- Cargar información de buses de singapur
3- Calcular componentes conectados
4- Establecer estación base:
5- Hay camino entre estacion base y estación:
6- Ruta de costo mínimo desde la estación base y estación:
7- Estación que sirve a mas rutas:
8- Cambiar metodo de busqueda ('bfs' o 'dfs'):
9- Existe un camino de busqueda entre base y estación:
10- Ruta de busqueda entre la estación base y estación:
0- Salir
*****
Seleccione una opción para continuar
>9
Estación destino (Ej: 15151-10): 15151-10
Hay camino de busqueda entre la estación base : y la estación: 15151-10:
True
```

```
*****
Seleccione una opción para continuar
>10
Estación destino (Ej: 15151-10): 15151-10
El camino de busqueda es de longitud: 36
75009-10
76059-10
76059-20
76069-20
76069-168
75059-168
75069-168
75349-168
84209-168
84529-168
84529-228
84219-228
84351-228
84009-228
84009-229
84009-26
84009-30
84029-30
84019-30
84019-30e
82069-30e
81199-30e
81189-30e
```

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

Existen diferencias porque bfs trabaja para calcular la ruta más corta de una sola fuente ya que usa el recuento mínimo desde un origen de vértice. DFS utiliza más bordes para atravesar hasta llegar al destino, lo que para nosotros era estación base y estación destino. Conociendo lo anterior por eso es que los caminos del dfs con respecto a los del bfs aumentaron (itera más veces). Ahora bien, las recomendaciones para el uso de ambos son las siguientes:

Bfs es adecuado para encontrar vértices dentro de áreas cercanas a la fuente de información (vertice inicial) y Dfs es adecuado para encontrar vértices alejados del vertice inicial. Con lo anterior, este laboratorio estaba diseñado para que su algoritmo más eficaz fuese bfs y en general aunque este es ineficiente en el consumo de memoria si es optimo para encontrar la distancia más corta.

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

Para el reto 4 se necesitan dos tipos de grafos: dirigido y no dirigido. En este laboratorio el grafo que trabajamos fue dirigido dado que las rutas tienen una dirección específica entre las estaciones, sin

embargo, en el reto tendremos diferentes rutas que apuntan al mismo destino, por lo cual necesitamos un grafo no dirigido.

La definición de lo anterior se da por:

“Los arcos de un grafo dirigido se representan gráficamente con flechas. Un grafo no dirigido es un grafo donde los arcos conectan a los nodos en ambos sentidos.”

Ejemplo en la vida cotidiana:

Grafo dirigido: Red de aguas de una ciudad puesto que cada tubería admite que el agua recorra en un solo sentido.

Grafo no dirigido: Carreteras, una carretera puede ser recorrida en ambos sentidos.

Ejemplo en nuestro Reto 4:

Grafo dirigido: Rutas que solo van a una dirección de destino y no tienen transbordo.

Grafo no dirigido: Rutas compartidas, que tienen la misma dirección de destino y tienen transbordo.

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

Los grafos serán conectados donde para cada par de vértices existe un camino entre ellos. Como se mencionó anteriormente habrá tanto dirigido como no dirigido. Los arcos en el grafo dirigido serán unidireccionales mientras que en el no dirigido será bidireccional.

En cuanto a la densidad que trata de el número de relaciones existentes entre el número de relaciones posibles mide y cuenta dichas relaciones o enlaces sobre el total. Por ejemplo, tenemos la red de los nodos de las rutas, nosotros queremos saber cuantas relaciones son (son los arcos/enlaces), lo que hace la densidad es que los cuenta y después se calcula el número total de nodos dividido los posibles.

Dado lo anterior todo parece indicar a que nuestro grafo no será disperso ya que el número de nodos no será tan bajo como si fuese un grafo vacío pero lo que sería optimo lograr es que se acorte a la mitad.

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

En el reto hablan sobre encontrar los caminos más cortos posibles desde un único vértice hasta todos los vértices del grafo, para ello se podría utilizar Dijkstra o Bellman-Ford.

Para encontrar caminos posibles bfs o dfs. Aún desconocemos si se utilizaría Prim o Kruskal, pero de los anteriores estamos seguros.