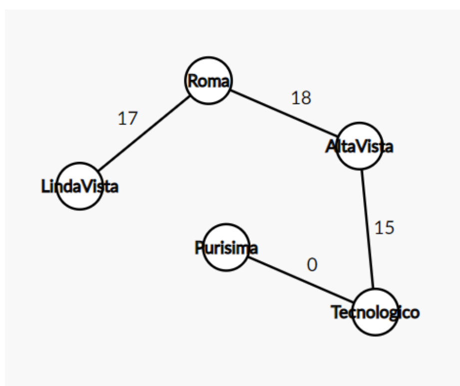
Patricio Santos A01197723

Actividad Integradora 2 – Reflexión

En la actividad integradora 2 hay 4 puntos que se deben resolver. La entrada que se da en la representación de un grafo con las ciudades de una región y sus conexiones de fibra óptica y su posición en un plano cartesiano. También se menciona si una ciudad es una central o no. Luego se dan conexiones de un nuevo cableado entre ciudades y futuras colonias que se quieren conectar. El primer punto requiere que se encuentre la forma óptima de conectar todo el nuevo cableado, tomando en cuenta las nuevas conexiones ya existentes, con el menor costo posible, de tal forma que se encuentren conectadas todas las colonias. El segundo punto requiere que se visiten todas las ciudades no centrales y se busca la ruta más corta en la que se pasa por cada ciudad solo una vez, aunque se pueden pasar por centrales también. El tercer punto pide que se encuentre el camino más optimo entre cada par existente de centrales y se puede pasar por no centrales. El cuarto y último punto pide que se encuentre el punto más cercano en el plano cartesiano donde se debería conectar cada colonia nueva.

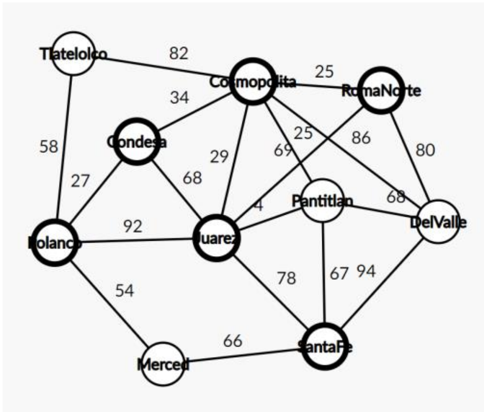
Para el primer punto se utilizó una adaptación del algoritmo de minimum spanning tree (MST). Se incluyeron todos los arcos del grafo de entrada con sus costos. Para cada nuevo cableado existente, se agrego el par de ciudades como un arco y se le puso un costo de 0, representando que la conexión ya existe y no incluye un costo extra. Se creo una estructura Colonia para guardar todos los datos de cada colonia y se les dio un índice en base a su orden de entrada para facilitar el uso de otros algoritmos. Con el primer caso de prueba que se da, la solución se puede representar de esta manera:



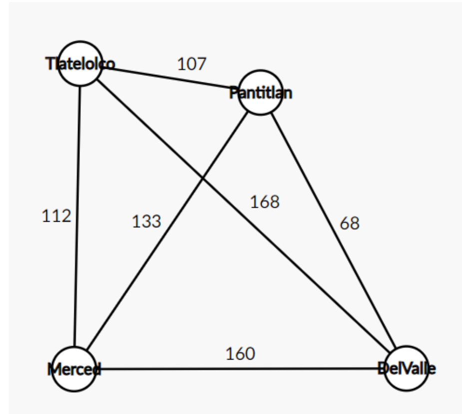
El arco con costo 0 representa una conexión nueva ya existente. Al final el costo total dio 50. El algoritmo de MST tiene una complejidad de O(n logn), n siendo los arcos del grafo. Esto se debe a que se pasa por todos los arcos mínimo una vez, y se van haciendo combinaciones (merge) al agrupar nodos ya conectados. Existe una llamada recursiva para encontrar los grupos de nodos.

Para el segundo punto se adaptó el algoritmo del viajero (TSP) para visitar todas las no centrales exactamente una vez. Para este punto se le tuvo que hacer una modificación al grafo de tal forma que solo hubiera no centrales, considerando los costos de caminos incluyendo centrales. La modificación de la tabla terminó con complejidad de $O(n^3)$ porque se visitan todas las conexiones de todas las colonias centrales para hacer los ajustes de conexiones. Después de estas modificaciones, el grafo del segundo caso de prueba quedó así:

Patricio Santos A01197723



-------->



Para determinar el costo de los arcos entre no centrales se tomó el costo más optimo entre las conexiones acumuladas entre centrales o caminos existentes con otras no centrales. Luego se aplicó el algoritmo TSP al grafo nuevo. Para encontrar el camino que se debía tomar se creó una matriz donde se guardaba el camino del costo acumulado. La matriz se iba cambiando de la siguiente manera:

	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	INF	Polanco	INF	INF	Juarez	INF	INF	Cosmopolita	INF
2	0	INF	0	INF	RomaNorte	Pantitlán	INF	INF	SantaFe	Cosmopolita	INF
3	0	Condesa	INF	0	INF	INF	Juarez	Tlatelolco	INF	INF	Merced
4	0	INF	DelValle	INF	0	INF	Juarez	INF	INF	Cosmopolita	INF
5	0	INF	DelValle	INF	INF	0	Juarez	INF	SantaFe	Cosmopolita	INF
6	0	Condesa	INF	Polanco	RomaNorte	Pantitlán	0	INF	SantaFe	Cosmopolita	INF
7	0	INF	INF	Polanco	INF	INF	INF	0	INF	Cosmopolita	INF
8	0	INF	DelValle	INF	INF	Pantitlán	Juarez	INF	0	INF	Merced
9	0	Condesa	DelValle	INF	RomaNorte	Pantitlán	Juarez	Tlatelolco	INF	0	INF
10	0	INF	INF	Polanco	INF	INF	INF	INF	SantaFe	INF	0

	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	INF	Polanco	INF	INF	Juarez	INF	INF	Cosmopolita	INF
2	0	INF	0	INF	RomaNorte	Pantitlán	INF	INF	SantaFe	Cosmopolita	INF
3	0	Condesa	INF	0	INF	INF	Juarez	Tlatelolco	INF	Condesa - Cosm	Merced
4	0	INF	DelValle	INF	0	INF	Juarez	INF	INF	Cosmopolita	INF
5	0	INF	DelValle	INF	INF	0	Juarez	INF	SantaFe	Cosmopolita	INF
6	0	Condesa	INF	Polanco	RomaNorte	Pantitlán	0	INF	SantaFe	Cosmopolita	INF
7	0	INF	INF	Polanco	INF	INF	INF	0	INF	Cosmopolita	INF
8	0	INF	DelValle	INF	INF	Pantitlán	Juarez	INF	0	INF	Merced
9	0	Condesa	DelValle	Condesa - Polan	RomaNorte	Pantitlán	Juarez	Tlatelolco	INF	0	INF
10	0	INF	INF	Polanco	INF	INF	INF	INF	SantaFe	INF	0

	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	INF	Polanco	INF	INF	Juarez	INF	INF	Cosmopolita	INF
2	0	INF	0	INF	RomaNorte	Pantitlán	INF	INF	SantaFe	Cosmopolita	INF
3	0	Condesa	INF	0	INF	INF	Juarez	Tlatelolco	INF	Condesa - Cosm	Merced
4	0	INF	DelValle	INF	0	INF	Juarez	INF	INF	Cosmopolita	INF
5	0	INF	DelValle	INF	INF	0	Juarez	INF	SantaFe	Cosmopolita	INF
6	0	Condesa	INF	Polanco	RomaNorte	Pantitlán	0	INF	SantaFe	Cosmopolita	Polanco - Merced
7	0	INF	INF	Polanco	INF	INF	INF	0	INF	Cosmopolita	Polanco - Merced
8	0	INF	DelValle	INF	INF	Pantitlán	Juarez	INF	0	INF	Merced
9	0	Condesa	DelValle	Condesa - Polan	RomaNorte	Pantitlán	Juarez	Tlatelolco	INF	0	Condesa - Polanc
10	0	INF	INF	Polanco	INF	INF	Polanco - Juarez	Polanco - Tlatelo	SantaFe	Polanco - Condes	0

Y al final se imprimía el camino de los costos acumulados. El algoritmo TSP tiene una complejidad de $O(2^n)$ porque se meten los nodos a una fila cuando son visitados como nodos hijo con costos óptimos. Al mismo tiempo cada nodo visita a todos sus hijos que no hayan sido aún visitados.

Para el tercer punto se utilizó el algoritmo de Floyd-Warshall. Primero se encontró el camino de todos los puntos a todos los puntos. Al mismo tiempo, se iba realizando una matriz auxiliar para guardar índices del camino que se puede tomar. El algoritmo tiene una complejidad de $O(n^3)$ porque se visitan todos los caminos hacia todos los nodos. En este caso no se tuvieron que realizar muchas adaptaciones a excepción de solo mostrar el camino de las centrales.

Patricio Santos A01197723

Para el cuarto y último punto no se tuvo que utilizar un algoritmo muy complejo como en los puntos pasados. Se utilizó un tipo de algoritmo voraz porque se estaba comparando la distancia de cada nueva colonia con todas las otras colonias existentes para encontrar aquella con la menor distancia en el plano cartesiano. Se obtuvo una complejidad de $O(n^2)$ por la explicación anterior.

Los algoritmos de los cuatro puntos se integraron de forma en la cual todos los resultados fueron impresos hacia un archivo de texto. En el programa principal (main) se iban guardando las respuestas de todos los algoritmos en una variable String. Hasta el final fue cuando se abrió un archivo de texto para escribir y se guardo toda la información de ese string en una sola línea. Todos los algoritmos comparten la estructura Colonia y el vector con los arcos. Estos se modifican dentro de cada algoritmo para sus resultados individuales. Los algoritmos que se usaron en esta actividad sirven mayormente para situaciones en las cuales se puede modelar un grafo. Una ventaja de estos es que se pueden utilizar en grafos con nodos de diferentes características, como en esta actividad que contenía ciudades centrales y no centrales. Una desventaja sería que, actualmente, la mayoría de los algoritmos para grafos no bajan de (n logn), y, por ejemplo, en el caso del algoritmo del viajero, entre más nodos y conexiones haya el tiempo que tarda el programa sube exponencialmente.