Humberto Genaro Cisneros Salinas

Reflexión

A lo largo de toda la actividad se utilizaron dos mapas globales para poder llevar una conversión directa en tiempo constante O(1) del nombre de la colonia a su índice en el grafo y vice versa.

Primero, se leen los datos de entrada del programa. Se crean las colonias bajo la estructura de la colonia con los atributos de nombre, coordenadas y si es central o no. Se agrega la referencia de su nombre e índice a los mapas globales. También se leen las conexiones entre las colonias y se agregan al grafo. Después se leen las nuevas conexiones y se agregan a la propiedad de built Edges del grafo. Por último, se leen las nuevas colonias y se agrega su referencia a los mapas globales.

Para el primer problema se identificó la necesidad de obtener el Minimum Spanning Tree del grafo, tomando en cuenta que ya existen conexiones construidas con el nuevo cableado. Para esto se utilizó el algoritmo de Kruskal, en donde se agregan las conexiones existentes antes de comenzar la ejecución. A lo largo de la ejecución se guarda el costo del MST y las conexiones identificadas como parte del mismo para poder desplegarlo como parte de la solución. La complejidad total de esta implementación es O(E*logE + E*logV) ya que E*logE es lo que toma ordenar las aristas y para cada arista se debe realizar la operación de unión con complejidad O(logV). Asumiendo que la cantidad de conexiones supera a la cantidad de colonias, la complejidad simplificada sería de O(E*logE).

Para el segundo problema se toma el algoritmo recursivo con bit masking para el problema del viajero. Se toma como caso base que el nodo ya haya sido visitado o que el nodo sea una central, en estos casos sale de la función. Si no, visita cada ciudad que no se ha visitado y establece el siguiente mejor nodo con la máscara y la posición establecida, guardando la ruta que se considera como óptima. Al final de la función de rutaOptima, se reconstruye la ruta óptima en un vector y se despliega junto con su costo. La implementación del problema del viajero tiene una complejidad temporal de O(2^n) al tener dos parámetros que cambian con cada llamada recursiva.

Para el tercer problema se busca generar la ruta óptima entre cada nodo que es central. Se utiliza el algoritmo de Floyd Warshall para encontrar el camino más corto entre todos los nodos del grafo, guardando los nodos por los que se pasa en cada combinación de origen-destino si son más eficientes que el anterior. Este algoritmo tiene una complejidad cúbica de O(n^3) por su triple ciclo for anidado. Una vez establecido la matriz de distancias más cortas y su auxiliar para los caminos, se despliegan aquellos que representan recorridos de una central a otra.

Finalmente, para el cuarto problema se busca el punto con la distancia más cercana a las colonias nuevas que se buscan agregar al grafo existente. Para esto se busca entre todas las colonias existentes la colonia con la distancia euclidiana más cercana a la colonia que se desea agregar y se despliega. Este algoritmo tiene una complejidad temporal de O(n*m) ya que para cada colonia nueva a agregar (m) se debe buscar entre todos las existentes en el plano (n).