Estudiante A: Andrés Felipe Gómez García, af.gomezg1@uniandes.edu.co, 202021189
Estudiante B: Tomás Bedoya Calixto, t.bedoya@uniandes.edu.co, 202020689

ANÁLISIS DE COMPLEJIDAD RETO 4

Requerimiento 1:

1. Se extrae el grafo de conexiones entre landing points y capitales y se realiza Kosaraju sobre este: O(E + V)
2. Se extrae el total de componentes a partir de lo retornado por Kosaraju: O(1)
3. Se extrae el id del landing point del mapa traductor que tiene de llave el nombre del landing point y de valor una lista donde el id está en la primera posición. Esto se hace con los dos landing points dados por el usuario: O(1)
4. Se extrae del diccionario landing_points el mapa de cables correspondientes a un lading point en específico, se realiza KeySet() y se extrae el cable de la primera posición para concatenarlo al landing point id. Esto se hace con los dos landing points dados por el usuario: O(C) donde c es el número de cables de un landing point.
5. Se utiliza stronglyConnected() para revisar si los dos landing points pertenecen al mismo componente fuertemente conectado: O(1)

Complejidad total: O(E + V)

Requerimiento 2:

1. Se crea una lista sencillamente encadenada para guardar el resultado: O(1)
2. Se realiza KeySet() del mapa landing_points: O(L) donde L es el número de landing points.
3. Por cada landing point, se saca el tamaño del mapa de cables que corresponde a dicho landing point y si este es estrictamente mayor a 1 se saca la información requerida de un mapa y se inserta en forma de tupla a la última posición de la lista de resultados: O(L)
4. Se imprimen los resultados con un loop que recorre la lista de resultados que incluye todos los landing points que sirven como puntos de interconexión a más cables en la red: O(L)

Complejidad Total: O(L)

Requerimiento 3:

1. Se extraen las capitales de los dos países dados del mapa_paises: O(1)
2. Se realiza Dijkstra sobre el grafo conexiones a partir de la capital 1: O(Elog(V))

3. Se extrae el costo de ir de capital 1 a capital 2: O(1)
4. Se utiliza pathTo() para extraer el camino de la capital 1 a la capital 2: O(V)
5. Se imprimen los resultados en un loop que se ejecuta por cada arco del camino: O(V)

Complejidad Total: O(Elog(V))

Requerimiento 4:

1. Se realiza eager prim sobre el grafo de conexiones empezando en el vertice Washington, D.C. debido a que se intuye que será el vertice desde el cual se obtendrá el árbol de mayor tamaño: O(Elog(V))
2. Se crea un grafo donde se guardará el mst y un mapa donde se guardarán los landing points para eliminar las repeticiones dado que un landing point puede tener varios vértices: O(1)
3. Se realiza KeySet() del mapa de vértices marked de la estructura search: O(V)
4. Por cada vertice, se realiza un split() para extraer el id del landing point y se inserta en el grafo del mst y en el mapa de landing points: O(V)
5. Se realiza KeySet() del mapa edgeTo y, por cada vertice en la lista resultante, se extrae el vertice de partida en el arco del diccionario edgeTo y el peso del arco y se añade el arco al grafo mst. También se van sumando los pesos para calcular el peso total del mst: O(E)
6. Se realiza DFS sobre el mst a partir de Washington, D.C. por lo que es la raíz del árbol: O(V + E)
7. Por cada vertice, se hace pathTo() y, en caso de que haya un camino, se revisa si el número de arcos es mayor al número máximo de arcos hasta el momento para ver si se asigna como el nuevo máximo. O(V!)
8. Se saca el número de arcos con numEdges() y se le suma 1 para sacar el número de vértices del gráfo que tienen conexiones. Esto por lo que en el mst se incluyeron todos los vértices: O(1)
9. Se imprimen los resultados en un loop que itera sobre la lista de arcos: O€

Complejidad General: O(V!)

Requerimiento 5:

1. Se extrae el id del landing point dado por el usuario del mapa traducción: O(1)
2. Se crea un mapa países para eliminar repeticiones de los países afectados y se inserta de una vez el país del landig point dado por el usuario: O(1)
3. Se realiza KeySet() del mapa de cables correspondientes al landing point: O(N)
4. Por cada cable del landing point, se buscan los vértices adyacentes al vertice id|cable y, por cada vértice adyacente, se extrae el país correspondiente y se inserta en el mapa de países para el resultado: O(CV)

5. Si el landing point es una capital, se buscan los vértices adyacentes a la capital y, por cada uno, se extrae el país correspondiente y se añade al mapa de países para el resultado: O(V)

Complejidad General: O(CV)

Medidas

	Carga de Datos	Req 1	Req 2	Req 3	Req 4	Req 5
Tiempo (ms)	4919.985	19231.467	1319.854	51298.603	3964.767	1143.008
Memoria (KB)	46109.916	133.441	128.559	28.879	149.293	15.412