

- Requerimiento 1.

Dado que el requerimiento solicita saber si el grafo cuenta con componentes fuertemente conectadas se debe aplicar el algoritmo de Kosaraju para poder identificar la cantidad claustrs conectados, adicionalmente este algoritmo permite saber si dos vertices están fuertemente conectados.

- Requerimiento 2.

Ya que se necesita saber cual Landing point es el que sirve como punto de interconexión a mas cables lo que se debe buscar es el vertice que tenga el grado mas alto. Para esto se obtiene una lista con todos los vertices del grafo y se debe ir recorriendo y preguntando el grado a cada vertice para así poder obtener el mayor.

- Requerimiento 3.

Este requerimiento se basa en encontrar caminos de costos mínimo en relación con la distancia en Km entre dos puntos. Para solucionar este requerimiento se debe aplicar Dijkstra o Bellman Ford para obtener las rutas de costo mínimo.

- Requerimiento 4.

Este requerimiento se basa en encontrar rutas de expansión mínima dentro de los landing points en relación con la distancia en Km entre estos, en este caso se puede aplicar cualquiera de los tres algoritmos que retornan el MST del grafo, pero se recomienda usar Kruskal dado que tiene una complejidad espacial menor respecto a los demás algoritmos.

- Requerimiento 5.

Este requerimiento se soluciona usando las funciones del TAD GRAFO presentes en la librería. Lo primero es obtener la lista de todos los cables adyacentes al grafo, lo cual está representado por los arcos del grafo. Luego se debe verificar a que pais pertenece el vertice de llegada del arco para poder obtener la lista de paises afectados, de esta manera, usando los arcos se puede obtener la distancia en Km que separa los vertices afectados para aplicar Merge SORT (ya que tiene la menor complejidad temporal entre los algoritmos de ordenamiento) y así, poder retornar lo solicitado.