Camilo Ortiz Cruz 201821615

Kevin Fernando Gomez

Requerimiento 1:

Texto

Descripción generada automáticamente

Como se puede ver la complejidad final es el número de vértices del grafo dirigido, esto se debe a que el grafo dirigido siempre va contener mas o los mismos vértices que el grafo no dirigido y por ende en notación O el termino mayor es el que queda, dado que este requerimiento era buscar un máximo consideramos que esta complejidad es muy buena ya que para encontrar el valor mas grande se debe recorrer todos los elementos por tanto consideramos que estamos cerca de los más optimo.

Requerimiento 2:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Como se puede observar la complejidad dio constante, esto se debe a que el calculo de los componentes fuertemente conectados (KosarajuSCC) se realizo en la carga de datos dado que este no necesita de ningún parámetro del usuario, dado que esta en el catalogo entonces las funciones necesarias para responder el requerimiento solo son connectedComponents y stronglyConnected y estas son constantes, consideramos que este es la solución óptima.

Requerimiento 3:

Texto

Descripción generada automáticamente

En el caso del requerimiento 3, la complejidad temporal la determina correr Dijkstra esto se debe a que el numero de arcos por el logaritmo de vertices es mayor a la complejidad de encontrar los aeropuertos, esto se debe a que en esencia la complejidad de encontrar el aeropuerto cercano es la de recorrer todos los aeropuertos que estén en el rango de búsqueda lo cual siempre será menor que todos los arcos del grafo multiplicado por el logaritmo del número de vertices.

Requerimiento 4:

Texto

Descripción generada automáticamente

Como se puede observar la complejidad del algoritmo está relacionada a realizar BFS, consideramos que este tiempo es razonable ya que para encontrar cual es el vertice que esta mas lejos de un origen ese necesario recorrer todos los vertices y arcos del arbol puesto que de otra forma no podríamos estar seguros y como podemos ver el resto de pasos del algoritmo solo tienen como complejidad un subconjunto de elementos de los arcos o los vertices, por lo cual E+V siempre será mayor.

Requerimiento 5:

Texto

Descripción generada automáticamente

El algoritmo tiene una complejidad que consideramos adecuada pues que para poder encontrar que aeropuertos son afectados se debe revisar todos los arcos entre el y todos los demás aeropuertos, sin embargo, dado que necesitamos saber los aeropuertos afectados entonces se debe revisar las listas de adyacencia de cada vertice del grafo y revisar si está el de interés, lo cual es exactamente lo que hace el algoritmo, algo importante a resaltar es que lo mas probable es que no se revisen las lista de adyacencia a totalidad si no que sean recorridos parciales pero su utiliza el peor caso posible.