

OBSERVACIONES DEL RETO 4

Estudiante 1 David A. Fuquen Flórez Cod 202021113. Requerimiento individual: 2
Estudiante 2 Juan Andrés Eslava Tovar Cod 202012035. Requerimiento individual: 3

Complejidad temporal de los requerimientos:

Requerimiento 1: La función “Top5Conectados” cuenta con la mayor complejidad de este requerimiento. En esta, se hace un recorrido por todos los vértices (aeropuertos) del digrafo `analyzer[“Di-aeropuertos”]` para poder conocer el indegree y outdegree de cada uno. Por ende, la complejidad es de $O(N)$, donde N es la cantidad de vértices en el grafo.

Requerimiento 2: La función “FindSCC” cuenta con la mayor complejidad de este requerimiento. En esta, se ejecuta el algoritmo de Korasaju para encontrar los elementos fuertemente conectados del digrafo `analyzer[“Di-aeropuertos”]`. Por ende, la complejidad será de $O(3(V+E))$, donde V es el número de vértices en el grafo y E es el número de arcos en el grafo.

Requerimiento 3: La función “RutaMenorCosto” cuenta con la mayor complejidad de este requerimiento. En esta, se ejecuta el algoritmo de Dijkstra para encontrar las rutas más baratas (en términos de distancia en kilómetros, es decir, más cortas) para el aeropuerto inicial en el digrafo `analyzer[“Di-aeropuertos”]`. Por ende, la complejidad será de $O(E^2 \log V)$, donde V es el número de vértices en el grafo y E es el número de arcos en el grafo.

Requerimiento 4: La función “RutaMasParadas” cuenta con la mayor complejidad temporal de este requerimiento. En esta, se ejecuta el algoritmo Prim para encontrar el MST del grafo no dirigido `analyzer[“NO-aeropuertos”]`. Por ende, la complejidad será de $O(E^2 \log V)$, donde V es el número de vértices en el grafo y E es el número de arcos en el grafo.

Requerimiento 5: La única función del requerimiento es “ChaoAirport”, por lo que su complejidad temporal será la del requerimiento. En esta, solo se ejecuta `adjacents` para conocer los vértices adyacentes al aeropuerto seleccionado por parámetro y `lt.size()` para conocer el tamaño de la lista de vértices adyacentes. Es decir, hay una orden de complejidad constante, $O(K)$.