Isai Daniel Chacón Silva – 201912015 - Correo: id.chacon@uniandes.edu.co

Nicolás Aparicio Claros – 201911357 – Correo: n.aparicioc@uniandes.edu.co

**Estructuras de datos y algoritmos**

**Reto 4**

En este documento se realiza el análisis de complejidad para cada uno de los algoritmos implementados para resolver cada requerimiento del reto 4 de la materia Estructuras de datos y algoritmos. En primer lugar, vale recalcar que el análisis teórico se realizó con la notación Big O, el cual es el peor caso posible para cada uno de los algoritmos realizados, sin embargo, este no ocurre siempre en la práctica y varía dependiendo en la distribución de las muestras de datos.

**Análisis de complejedidad**

**Requerimiento 1**

La complejidad de este requerimiento viene dada principalmente por la función *interconexión(catalog)*. En esta se recorre la lista de vértices del grafo dirigido O(V) para formar un árbol binario de conexiones, cuyo *key* sea la suma de los términos *indegree* y *outdegree*. En caso de que una *key* se repita, se anexa al *value*, pues esta se encuentra definida como un ArrayList. Una vez se construya el mapa ordenado se itera sobre el valor obteniendo su máximo, lo cual es sencillo pues el árbol se encuentra ordenado O(logV), y se añade a una nueva lista hasta que esta posea los 5 elementos requeridos en el *view.* De modo que en total se tiene una complejidad de O(V).

**Requerimiento 2**

Para este requerimiento se llama a la función *clusters(catalog, iata1, iata2)*. Esta función corre el algoritmo de Kosaraju para obtener los componentes conexos del digrafo. Luego, simplemente se retornan los componentes conexos del grafo y si para 2 iatas que entran por parámetro una variable booleana que indica si se encuentran dentro de un mismo clúster. Este requerimiento posee la misma complejidad temporal de Kosaraju, la cual es O(V+E) ya que se tiene una representación del grafo bajo listas de adyacencia.

**Requerimiento 3**

**Requerimiento 4**

**Requerimiento 5**