

## **Análisis de Complejidad Requerimientos Reto 4**

### **Integrantes:**

Alejandro Herrera Jiménez, 201915788

Hugo David Flórez Rodríguez, 201912429

### **Carga de datos**

En la carga de datos se implementaron diversas funciones. Por una parte, se cargaron 3 árboles distintos y por otra se cargó un grafo. Para el caso de los árboles cada carga tenía una complejidad de  $O(N)$ , mientras que para el grafo se tenía una complejidad de orden  $O(N)$ . Por lo tanto, la complejidad total es de  $O(N)$ .

### **Requerimiento 1**

En el requerimiento 1 se implementó un algoritmo que recorría en su totalidad todos los nodos del grafo para revisar cuantos arcos asociados poseía cada uno, lo que se considera es de orden  $O(N)$ . Se hacía este mismo procedimiento 5 veces guardando y eliminando paulatinamente el máximo valor hasta obtener los 5 mayores, lo que se considera de orden  $O(5N)$  simplificando. Por lo tanto, la complejidad total del algoritmo es de  $O(N)$ .

### **Requerimiento 2**

Para el requerimiento 2 primero se aplicaba el algoritmo de Kosaraju para encontrar los componente fuertemente conectados, el cual tiene una complejidad temporal de  $O(N)$ . Luego, revisa si los dos aeropuertos dados de entrada están o no fuertemente conectados, lo cual tiene complejidad en el peor de los casos de  $O(\# \text{ de Componentes conectados})$ . Siendo  $\# \text{ de Componentes conectados}$  menor a  $N$ . Por lo tanto, la complejidad del requerimiento es de  $O(N)$ .

### **Requerimiento 3**

En el requerimiento 3 se usó el algoritmo de Dijkstra, el cual posee complejidad de  $O(N \log(N))$ . Luego, se recorrió la pila obtenida por el algoritmo para obtener la información de cada aeropuerto, lo cual no representa una complejidad de más dos cifras en la casi todos los casos. Después, para imprimir todos los datos solicitados de cada aeropuerto se recorría 5 veces el árbol de los datos, lo cual posee complejidad de búsqueda de  $O(5 * 1.39 \log(N))$  por ser BST's, es decir  $O(6.95 \log(N))$ . Es por esto que la complejidad total del algoritmo es  $O(N \log(N))$ .

#### **Requerimiento 4**

Para este requerimiento la complejidad es la misma del requerimiento 3 solo que  $N$  veces. Por lo que la complejidad total sigue siendo de  $O(N^2 \log(N))$ .

#### **Requerimiento 5**

El orden de este requerimiento es de  $O(N \log(N))$  debido a que debe buscar un vértice en el grafo. Luego se asoció dicho vértice con sus vértices adyacentes, lo cual no representa una mayor complejidad. Por lo que la complejidad total del algoritmo es  $O(N \log(N))$ .