

Documento de análisis | Reto 4

Integrantes:

Sergio Franco, 202116614, s.francop@uniandes.edu.co

Requerimiento 1

$O(V \log V)$

La complejidad se debe a este fragmento de código de ordenamiento, ya que se usa el algoritmo shellsort.

```
data = sortByConnections(data)
```

Requerimiento 2

$O(V + E)$

La complejidad se debe a este fragmento de código, ya que se usa el algoritmo Kosara para encontrar los componentes conectados.

```
components = scc.KosarajuSCC(graph)
```

Requerimiento 3

$O(E \log V)$

La complejidad se debe a este fragmento de código, ya que se usa el algoritmo Dijkstra para encontrar los caminos en el grafo.

```
analyzer['paths'] = djik.Dijkstra(  
    analyzer["directedGraph"],  
    initialAirport  
)
```

Requerimiento 5

$O(E)$

La complejidad se debe a este fragmento de código, ya que el método debe recorrer cada arco relacionado al vértice.

```
affected = gr.adjacents(graph, IATA)
```

Requerimiento 6

$O(E \log V)$

La complejidad se debe a este fragmento de código, ya que se usa el algoritmo Dijkstra para encontrar los caminos en el grafo.

```
analyzer['paths'] = djik.Dijkstra(  
    analyzer["directedGraph"],  
    initialAirport  
)
```