**ANÁLISIS DEL RETO**

Sergio Alberto Ramirez Lara, 202211388, sa.ramirezl1@uniandes.edu.co

# **Requerimiento <<1>>**

## **Descripción**

Este requerimiento recibe como parámetro de entrada el punto de origen y el punto de destino. Si los puntos de origen están ubicados a mas de 30Km del aeropuerto el algoritmo no se ejecuta y indica el aeropuerto mas cercano al punto ingresado.

|  |  |
| --- | --- |
| **Entrada** | El punto de origen (una localización geográfica con latitud y longitud)  El punto de destino (una localización geográfica con latitud y longitud) |
| **Salidas** | Si el algoritmo se ejecuta la respuesta contiene la distancia total que tomará el camino entre el punto de origen y el de destino, el número de aeropuertos que se visitan en el trayecto y el camino encontrado. |
| **Implementado (Sí/No)** | Si |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

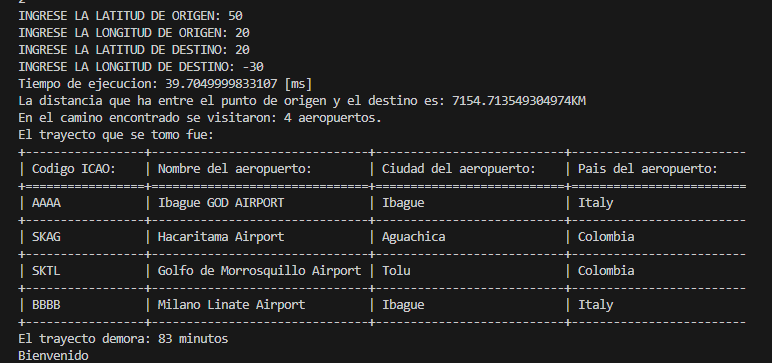
|  |  |
| --- | --- |
| **Pasos** | **Complejidad** |
| Verifica si la distancia de origen y de destino esta a menos de 30 Km de algún aeropuerto | O(V) |
| Si existen los aeropuertos se crean los caminos desde el origen hacia el aeropuerto destino | O((V+E) log(V)) |
| Se hallan los caminos en los grafos con los pesos de distancia y tiempo | O(1) |
| Se saca de la cola del camino el ultimo elemento para sumar la distancia entre los aeropuertos, el tiempo de vuelo y las paradas | O(1) |
| De la lista de los aeropuertos se saca la información de las paradas para agregarlos a un mapa para obtener su información. | O(V) |
| ***TOTAL*** | ***O(***((V+E) log(V)***)*** |

## **Pruebas Realizadas**

Se ejecuto el algoritmo para hallar el camino entre dos aeropuertos. (Para algunas pruebas se crearon aeropuertos y vuelos ficticios)

Prueba 1

|  |  |
| --- | --- |
| **Entrada** | **Tiempo (s)** |
| Latitud de origen | 50 |
| Longitud de origen | 20 |
| Latitud de destino | 20 |
| Longitud de destino | -30 |



Prueba 2

|  |  |
| --- | --- |
| **Entrada** | **Tiempo (s)** |
| Latitud de origen | 50 |
| Longitud de origen | 20 |
| Latitud de destino | 7.5 |
| Longitud de destino | -73.4 |

Texto

Descripción generada automáticamente

# **Requerimiento <<3>>**

## **Texto Descripción generada automáticamente**

## **Descripción**

Este requerimiento no recibe ningún parámetro, sino que al ejecutarse calcula un grafo con origen en el aeropuerto con mayor concurrencia comercial y calcula los trayectos hacia la mayor cantidad de aeropuertos con la menor distancia recorrida.

|  |  |
| --- | --- |
| **Entrada** | N/A |
| **Salidas** | La respuesta contiene el aeropuerto con mayor concurrencia comercial, los trayectos a cada aeropuerto, con su distancia y tiempo y la distancia total que se recorren en todos los trayectos |
| **Implementado (Sí/No)** | Si |

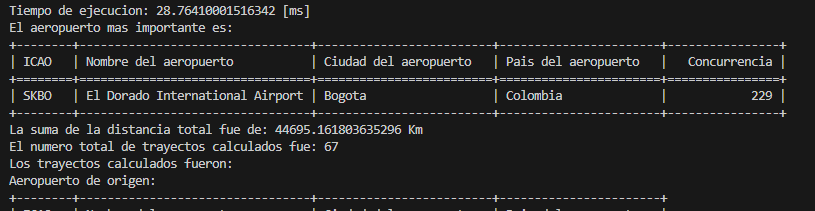
## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

|  |  |
| --- | --- |
| **Pasos** | **Complejidad** |
| De la estructura de datos que tiene la concurrencia saca el aeropuerto con mayor concurrencia comercial | O(1) |
| Halla el grafo con el algoritmo Dijkstra de el tiempo y la distancia | O((V+E) log(V)) |
| Halla si tiene camino con cada Aeropuerto y de ser va guardando en un diccionario, su origen, su destino y cuanto demora la ruta además de que distancia recorre | O(V) |
| ***TOTAL*** | ***O(***((V+E) log(V)***)*** |

## **Pruebas Realizadas**

Se ejecuto el algoritmo para hallar el camino entre dos aeropuertos. (Para algunas pruebas se crearon aeropuertos y vuelos ficticios)



Por cada trayecto viene la siguiente información:

Captura de pantalla de computadora

Descripción generada automáticamente

# **Requerimiento <<4>>**

## **Texto Descripción generada automáticamente**

## **Descripción**

Este requerimiento no recibe ningún parámetro, sino que al ejecutarse calcula un grafo con origen en el aeropuerto con mayor importancia de carga y calcula los trayectos hacia la mayor cantidad de aeropuertos con la menor distancia recorrida.

|  |  |
| --- | --- |
| **Entrada** | N/A |
| **Salidas** | La respuesta contiene el aeropuerto con mayor importancia de carga, los trayectos a cada aeropuerto, con su distancia, tipos de aeronaves de los trayectos, tiempo y la distancia total que se recorren en todos los trayectos |
| **Implementado (Sí/No)** | Si |

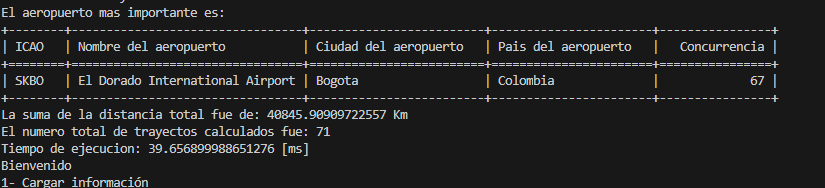
## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

|  |  |
| --- | --- |
| **Pasos** | **Complejidad** |
| De la estructura de datos que tiene la concurrencia saca el aeropuerto con mayor importancia de carga | O(1) |
| Halla el grafo con el algoritmo Dijkstra del tiempo y la distancia | O((V+E) log(V)) |
| Halla si tiene camino con cada Aeropuerto y de ser va guardando en un diccionario, su origen, su destino y cuanto demora la ruta además de que distancia recorre | O(V) |
| ***TOTAL*** | ***O(***((V+E) log(V)***)*** |

## **Pruebas Realizadas**

Se ejecuto el algoritmo para hallar el camino entre dos aeropuertos. (Para algunas pruebas se crearon aeropuertos y vuelos ficticios)



Por cada trayecto viene la siguiente información:

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

# **Requerimiento <<6>>**

## **Texto Descripción generada automáticamente**

## **Descripción**

Este requerimiento recibe como parámetro el número de aeropuertos que se desea cubrir. Como resultado da el aeropuerto con mayor importancia comercial, además de los trayectos a los n aeropuertos mas importantes que le siguen, cada trayecto con los aeropuertos incluidos en el camino y la distancia en Km del trayecto.

|  |  |
| --- | --- |
| **Entrada** | N/A |
| **Salidas** | La respuesta contiene el aeropuerto con mayor importancia comercial, además de los trayectos a los n aeropuertos más importantes que le siguen, cada trayecto con los aeropuertos incluidos en el camino, los vuelos del camino y la distancia en Km del trayecto |
| **Implementado (Sí/No)** | Si |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

|  |  |
| --- | --- |
| **Pasos** | **Complejidad** |
| De la estructura de datos que tiene la concurrencia saca el aeropuerto con mayor importancia comercial | O(1) |
| Halla el grafo con el algoritmo Dijkstra del grafo de los vuelos comerciales con las distancias entre sus aeropuertos | O((V+E) log(V)) |
| Hallar los n aeropuertos que le siguen en importancia al aeropuerto origen | O(n) |
| Para cada n aeropuerto hallar si tiene camino con el aeropuerto original | O(n) |
| Hallar el camino hacia ese aeropuerto, guardando en un diccionario los datos de cada aeropuerto que se visita en el trayecto, además de los vuelos que se visitaron | O(1) |
| ***TOTAL*** | ***O(***((V+E) log(V)***)*** |

## **Pruebas Realizadas**

Se ejecuto el algoritmo para hallar el camino entre dos aeropuertos. (Para algunas pruebas se crearon aeropuertos y vuelos ficticios)



# **Requerimiento <<7>>**

## **Texto Descripción generada automáticamente**

## **Descripción**

Este requerimiento recibe como parámetro un punto de origen y un punto de destino, y lo queda como resultado es el camino más corto en cuestión de tiempo entre los aeropuertos que queden a menos de 30 Km entre los puntos dados

|  |  |
| --- | --- |
| **Entrada** | N/A |
| **Salidas** | De ejecutarse el algoritmo la respuesta tiene los aeropuertos de origen y destino que queden a menos de 30Km de los parámetros ingresados, de no ejecutarse el algoritmo da como resultado cual es el aeropuerto mas cercano a los parametros ingresados |
| **Implementado (Sí/No)** | Si |

## **Análisis de complejidad**

Análisis de complejidad de cada uno de los pasos del algoritmo

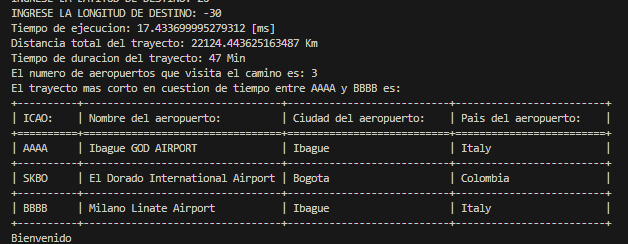
|  |  |
| --- | --- |
| **Pasos** | **Complejidad** |
| Verifica si la distancia de origen y de destino está a menos de 30 Km de algún aeropuerto | O(V) |
| Si existen los aeropuertos se crean los caminos desde el origen hacia el aeropuerto destino en el grafo de los vuelos comerciales con el peso de tiempo de vuelo | O((V+E) log(V)) |
| Si existe un camino se halla ese camino, guardando en un diccionario por cada aeropuerto visitado, la distancia entre estos, el tiempo del recorrido y los datos del aeropuerto | O(1) |
| De no existir los aeropuertos a menos de 30 Km se halla los aeropuertos mas cercanos a los parámetros de entrada y se da como resultado | O(1) |
| ***TOTAL*** | ***O(***((V+E) log(V)***)*** |

## **Pruebas Realizadas**

Se ejecuto el algoritmo para hallar el camino entre dos aeropuertos. (Para algunas pruebas se crearon aeropuertos y vuelos ficticios)

Prueba 1

|  |  |
| --- | --- |
| **Entrada** | **Tiempo (s)** |
| Latitud de origen | 50 |
| Longitud de origen | 20 |
| Latitud de destino | 20 |
| Longitud de destino | -30 |



Prueba 2

|  |  |
| --- | --- |
| **Entrada** | **Tiempo (s)** |
| Latitud de origen | 50 |
| Longitud de origen | 20 |
| Latitud de destino | 7.5 |
| Longitud de destino | -73.4 |

