Documento Análisis

Estructura de Datos y Algoritmos

Reto 4

**Integrantes:**

Sofía Escobar Tamayo - [s.escobart@uniandes.edu.co](mailto:s.escobart@uniandes.edu.co)

Juan Manuel Jauregui Rozo -  [j.jauregui@uniandes.edu.co](mailto:j.jauregui@uniandes.edu.co)

**Análisis de complejidad de cada requerimiento:**

**Requerimiento 1:** Este requerimiento cuenta con una sola función, Requerimiento1, que se encarga de resolver todo el problema. Cuenta con dos ciclos de la misma jerarquía que generan una complejidad lineal, es decir, O(N). Como no hay ordenamientos ni algoritmos de grafos y solo hay operaciones que se realizan en O(k) dentro del ciclo se puede concluir que la complejidad temporal de este requerimiento es O(N). A continuación se muestra el tiempo de la carga de datos y del requerimiento en tabla y gráfica de acuerdo con el tamaño del archivo utilizado.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Archivo** | **Datos Rutas** | **Datos Aeropuertos** | **Tiempo Requerimiento** | **Tiempo Carga** |
| small | 39 | 181 | 15,62 | 2937,5 |
| 5pct | 253 | 453 | 15,62 | 2953,12 |
| 10pct | 1157 | 907 | 31,25 | 3343,75 |
| 20pct | 4923 | 1815 | 62,50 | 3687,5 |
| 30pct | 11781 | 2722 | 93,75 | 4796,88 |
| 50pct | 28557 | 4537 | 125,00 | 9734,38 |
| 80pct | 62867 | 7260 | 140,62 | 21250 |
| large | 92593 | 9075 | 343,75 | 40578,12 |

**Requerimiento 2:** Este requerimiento cuenta solamente con una función, Requerimiento2, que se encarga de resolver todo el problema. En la función se utiliza el algoritmo de Kosaraju para hallar el número de componentes conectados. Como este algoritmo es lo único que se realiza en la función se puede decir que la complejidad temporal es de O(V+E) donde V es el número de vértices del grafo y E corresponde al número de arcos del grafo. A continuación se muestra el tiempo de la carga de datos y del requerimiento en tabla y gráfica de acuerdo con el tamaño del archivo utilizado.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Archivo** | **Datos Rutas** | **Datos Aeropuertos** | **Tiempo Requerimiento** | **Tiempo Carga** |
| small | 39 | 181 | 31,25 | 2937,5 |
| 5pct | 253 | 453 | 78,12 | 2953,12 |
| 10pct | 1157 | 907 | 468,75 | 3343,75 |
| 20pct | 4923 | 1815 | 562,50 | 3687,5 |
| 30pct | 11781 | 2722 | 1312,50 | 4796,88 |
| 50pct | 28557 | 4537 | 2843,75 | 9734,38 |
| 80pct | 62867 | 7260 | 6031,25 | 21250 |
| large | 92593 | 9075 | 7984,38 | 40578,12 |

**Requerimiento 3:**

**Requerimiento 4:** Este requerimiento cuenta con 2 funciones. La principal, Requerimiento4, resuelve la mayoría del problema, mientras que una función secundaria, getDataIATAList2, se encarga de extraer información del archivo de rutas a partir de códigos IATA. En la función principal se encuentran dos ciclos con la misma jerarquía, el algoritmo DFS y el algoritmo Prim. De todas estas opciones la complejidad temporal más alta corresponde al algoritmo Prim. Este algoritmo cuenta con una complejidad temporal O(ELog(V)) donde E corresponde al número de arcos del grafo y V corresponde al número de vértices del grafo. A continuación se muestra el tiempo de la carga de datos y del requerimiento en tabla y gráfica de acuerdo con el tamaño del archivo utilizado.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Archivo** | **Datos Rutas** | **Datos Aeropuertos** | **Tiempo Requerimiento** | **Tiempo Carga** |
| small | 39 | 181 | 62,50 | 2937,5 |
| 5pct | 253 | 453 | 250,00 | 2953,12 |
| 10pct | 1157 | 907 | 609,38 | 3343,75 |
| 20pct | 4923 | 1815 | 1609,38 | 3687,5 |
| 30pct | 11781 | 2722 | 2765,62 | 4796,88 |
| 50pct | 28557 | 4537 | 9500,00 | 9734,38 |
| 80pct | 62867 | 7260 | 14781,25 | 21250 |
| large | 92593 | 9075 | 32500,00 | 40578,12 |

**Requerimiento 5:** Este requerimiento cuenta con 2 funciones. La primera es Requerimiento 5, la cual no contiene ciclos importantes, solamente tiene dos que son irrelevantes ya que son para crear las listas de primeros y últimos 3. Por otro lado, se tiene otra función que se llama GetDataIATAList la cual tiene dos ciclos pero uno de ellos contiene muy pocos datos (los adyacentes a un vértice) por lo cual no genera una complejidad O(N^2). Aun así, para que el print quede en orden alfabético se debe realizar un sort con el algoritmo mergesort. Como no hay nada mayor en términos de notación Big O, la complejidad del requerimiento sería la misma complejidad del algoritmo de ordenamiento, es decir, O(NLog(N)). Como se puede ver en la gráfica, el tiempo que se demora la carga de datos aumenta de manera exponencial a medida que se van añadiendo datos mientras que el requerimiento siempre es instantáneo (0 ms) sin importar el tamaño del archivo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Archivo** | **Datos Rutas** | **Datos Aeropuertos** | **Tiempo Requerimiento** | **Tiempo Carga** |
| small | 39 | 181 | 0,00 | 2937,5 |
| 5pct | 253 | 453 | 0,00 | 2953,12 |
| 10pct | 1157 | 907 | 0,00 | 3343,75 |
| 20pct | 4923 | 1815 | 0,00 | 3687,5 |
| 30pct | 11781 | 2722 | 0,00 | 4796,88 |
| 50pct | 28557 | 4537 | 0,00 | 9734,38 |
| 80pct | 62867 | 7260 | 0,00 | 21250 |
| large | 92593 | 9075 | 0,00 | 40578,12 |