

Noviembre 2020

Estructuras De Datos Y Algoritmos

Reto 4

Complejidad Temporal Algoritmos Implementados

En general, se utilizó un mapa para guardar los datos necesarios de cada vértice (estación). Tanto para el requerimiento 3, como para el 6, se hace uso de listas que ordenan estos vértices en función de algún valor.

Requerimiento 3: Estaciones Críticas.

Para la solución de este requerimiento, se implementó una función que crea tres listas de estaciones y utiliza tres veces el método de ordenamiento MergeSort sobre ellas. Para almacenar los datos en listas, se recorren todas las llaves del mapa. La primera lista queda ordenada en función del número de viajes que llegan a la estación, la segunda en función del número de viajes que salen de la estación, y la tercera en función de la suma de los dos datos anteriores.

Por lo tanto, la complejidad en tiempo del algoritmo es de: $O(N + 3(N \log N))$

Requerimiento 5: Recomendador de Rutas.

Para la solución de este requerimiento, se implementó un algoritmo que busca en el mapa, las estaciones de las que salen más viajes y a las que llegan más viajes dado un rango de edad específico. A continuación, se aplica el algoritmo Dijkstra para conocer la ruta entre estas dos estaciones y el costo mínimo de esta.

Por lo tanto, la complejidad en tiempo del algoritmo es de: $O(V + E \log V)$

Siendo E el número de arcos, y V el número de vértices

Requerimiento 6: Ruta de Interés Turístico

Se crean dos listas de estaciones. La primera lista será ordenada en función de la cercanía con la coordenada inicial ingresada, y la segunda será ordenada según la cercanía con la coordenada final ingresada. Se recorre todas las llaves del mapa para que con cada llave (estación/vértice) se calcule la distancia harvesiana a las coordenadas dadas. Para ordenar las listas, se hace uso del algoritmo MergeSort. Cuando se hallan las estaciones más cercanas a los puntos ingresados, se hace uso del algoritmo Dijkstra para conocer la ruta entre estas dos estaciones y el costo mínimo de esta.

La complejidad en tiempo del algoritmo es de: $O(V + 2(V \log V) + V + E \log V)$