Daniel Rincón G.

Nikolas Castellanos B.

Diseño- Proyecto 2

***Requerimientos:***

**Parte A (Nikolas Castellanos B.)**

|  |  |
| --- | --- |
| **Nombre** | 1A. Obtener las N letras más frecuentes por las que comienza el nombre de una zona (No diferenciar las mayúsculas de las minúsculas). N es un dato de entrada. El resultado debe aparecer de mayor a menor. Para cada letra se debe imprimir la letra y el nombre de las zonas que comienzan por esa letra. |
| **Resumen** | Este método recorre los datos que van a ser cargados al proyecto, particularmente el ID con el que se identifica cada zona y posteriormente se compara con el ID de una siguiente para obtener las letras más recurrentes en los identificadores de las zonas, el criterio de número de letras más frecuentes es un dato de entrada, es decir, que el usuario especifica cuantas letras de recurrencia desea consultar y el resultado se muestra ordenado lexicográficamente de forma descendente. Para este método se ignora el uso de la función compareToIgnoreCase |
| **Entradas** | |
| (N): El número de letras que se quiere consultar que tengan mayor frecuencia de repetición al comienzo de los nombres de las zonas | |
| **Resultados** | |
| Se imprime cada letra por la cual comienza el nombre de una zona y todos los nombres de las zonas cuyo nombre empiece por dicha letra. Los datos se imprimen de forma ordenada | |
| **Estimación de complejidad temporal** | |
| Teniendo en cuenta que los datos van a estar ordenados, su complejidad temporal dependerá del algoritmo de ordenamiento que se use teniendo así una complejidad entre O(N log N) para el mejor caso, o una complejidad de O(N2) la complejidad temporal por ordenamiento, se puede decir que es despreciable, puesto que algunas estructuras contienen los datos ordenados o parcialmente ordenados. Adicionalmente este método debe considerar las ocurrencias para cada letra por lo que se requiere hacer un recorrido sobre la estructura en la cual se encuentren los datos se podría tener una complejidad entre O(N)para búsqueda y O(N2) en caso de que se haga un ciclo anidado para realizar comparaciones | |

|  |  |
| --- | --- |
| **Nombre** | 2A. Buscar los nodos que delimitan las zonas por Localización Geográfica (latitud, longitud). |
| **Resumen** | Este método permite consultar los datos delimitados según la latitud y longitud dada. Todos los nodos que se encuentren en la frontera de las zonas, se debe truncar las primeras tres cifras decimales. |
| **Entradas** | |
| Latitud para a consulta de zonas | |
| Longitud para a consulta de zonas | |
| **Resultados** | |
| El número de nodos que se encuentran delimitando las zonas | |
| Latitud de cada nodo encontrado | |
| Longitud de cada nodo encontrado | |
| Nombre de la zona de acuerdo al nodo encontrado | |
| **Estimación de complejidad temporal** | |
| Este método puede tener una complejidad de O (log N) o O(K) dependiendo de la estructura que se use para hacer la búsqueda del requerimiento | |

|  |  |
| --- | --- |
| **Nombre** | 3A. Buscar los tiempos promedio de viaje que están en un rango y que son del primer trimestre del 2018. |
| **Resumen** | Para este requerimiento se tiene un rango de tiempos dado para realizar la búsqueda de todos los viajes cuyo tiempo promedio mensual esté en ese rango |
| **Entradas** | |
| **[limite bajo, limite\_alto]:** **limete\_bajo:** Es el punto inferior del intervalo de los tiempos promedio dados por parámetro para realizar la comparación. | |
| **[limite\_bajo, limite\_alto]:** **limite\_alto:** Es el punto superior del intervalo de zonas dadas por parámetro para realizar la comparación. | |
| **Resultados** | |
| Los N viajes ordenados por zona origen y zona destino | |
| Zona origen para cada viaje dentro del rango | |
| Zona destino para cada viaje dentro del rango | |
| Mes para cada viaje dentro del rango | |
| Tiempo promedio de viaje mensual para cada viaje dentro del rango | |
| **Estimación de complejidad temporal** | |
| Para este requerimiento se debe hacer una búsqueda y un ordenamiento y dependiendo de la estructura que se use, la complejidad puede ser distinta. Se espera que este método tenga una complejidad entre O (log n) + O (n2) y O (n)/O(K) + O(nlog n). | |

**Parte B (Daniel Rincón G.)**

|  |  |
| --- | --- |
| **Nombre** | 1B. Buscar los N zonas que están más al norte |
| **Resumen** | Buscar las N zonas más al norte del mapa. Una zona está más al norte que otra si la latitud de la zona A es mayor que la latitud de la zona B. |
| **Entradas** | |
| La cantidad de zonas que se quieren saber | |
| **Resultados** | |
| Una estructura de datos con las N zonas más al norte ordenados de más al norte a más sur | |
| Por cada zona de la estructura se debe imprimir el nombre de la zona, su latitud y longitud. | |
| **Estimación de complejidad temporal** | |
| En este método toca encontrar las N zonas más al norte. Entonces para este vamos a usar el ordenamiento quick sort. Este ordenamiento tiene una complejidad de NlogN. Por lo tanto, el método tendría una complejidad de NlogN ya que entre todos los funcionamientos es el de mayor tiempo. | |

|  |  |
| --- | --- |
| **Nombre** | 2B. Buscar nodos de la malla vial por Localización Geográfica |
| **Resumen** | Dado una latitud y una longitud, se deben mostrar todos los nodos que tengan esas mismas latitud y longitud truncando a 2 cifras decimales. |
| **Entradas** | |
| Latitud deseada | |
| Longitud deseada | |
| **Resultados** | |
| Una estructura con todos los nodos que cumplan los requisitos | |
| Por cada nodo se debe imprimir su id, su latitud y su longitud | |
| **Estimación de complejidad temporal** | |
| La estimación de complejidad temporal de este método es de aproximadamente N. Teniendo en cuenta que los datos están ordenados y toca recorrer toda la estructura para ver si cada nodo cumple con los requisitos, la complejidad del algoritmo sería N. | |

|  |  |
| --- | --- |
| **Nombre** | 3B. Buscar los tiempos de espera que tienen una desviación estándar en un rango dado y que son del primer trimestre del 2018. |
| **Resumen** | Dado un rango de desviaciones estándares [limite\_bajo, limite\_alto] retornar los viajes cuya desviación estándar mensual este en ese rango. |
| **Entradas** | |
| N viajes deseados | |
| Desviación estandar abajo | |
| Desviación estandar alto | |
| **Resultados** | |
| Los N viajes ordenados por zona de origen y zona de destino | |
| De cada viaje se debe mostrar: zona origen, zona destino, mes y la desviación estandar | |
| **Estimación de complejidad temporal** | |
| Como el resultado debe quedar ordenado vamos a usar el ordenamiento quicksort. Este tiene una complejidad de NlogN. Vamos a usar este ordenamiento para poder hacer el recorrido hasta los N datos y no tener que buscar los N datos en toda la estructura original (sin ordenar). | |

**Parte C**

|  |  |
| --- | --- |
| **Nombre** | 1C. Retornar todos los tiempos de viaje promedio que salen de una zona dada y a una hora dada. |
| **Resumen** | Se recibe el id de una zona de destino y una hora ingresadas por el usuario para realizar la búsqueda de los tiempos promedios que cumplen con los criterios de búsqueda. |
| **Entradas** | |
| ID zona destino | |
| Hora | |
| **Resultados** | |
| Zona Origen de cada viaje que cumple con las condiciones de búsqueda | |
| Zona Destino de cada viaje que cumple con las condiciones de búsqueda | |
| Hora de cada viaje que cumple con las condiciones de búsqueda | |
| Tiempo promedio de cada viaje que cumple con las condiciones de búsqueda | |
| **Estimación de complejidad temporal** | |
| Se espera que para este requerimiento, la complejidad en el peor de los casos sea de O(N) y en el mejor de los casos O (n log n). | |

|  |  |
| --- | --- |
| **Nombre** | 2C. Retornar todos los tiempos de viaje que llegan de una zona dada y en un rango de horas. |
| **Resumen** | Se recibe el ID de una zona de llegada y un rango de horas ingresadas por el usuario para mostrar los tiempos promedio de viaje que cumplan con los criterios de búsqueda. |
| **Entradas** | |
| Zona destino, criterio de búsqueda | |
| **[hora\_min, hora\_max]:** Rango de horas definido por e usuario | |
| **Resultados** | |
| Tiempo promedio de cada viaje que cumpla con los criterios de búsqueda | |
| La zona origen de cada viaje. | |
| La zona destino de cada viaje. | |
| Hora de cada viaje. | |
| **Estimación de complejidad temporal** | |
| Se espera que para este requerimiento al realizar una búsqueda sobre los datos contenidos en cierta estructura, la complejidad en el peor de los casos sea de O(N) y en el mejor de los casos O (n log n). La complejidad varía dependiendo de la estructura en la que se encuentren los datos | |

|  |  |
| --- | --- |
| **Nombre** | 3C. Obtener las N zonas priorizadas por la mayor cantidad de nodos que definen su frontera. |
| **Resumen** | Para este requerimiento se debe consultar la cantidad de zonas (N zonas) de los viajes que se encuentran en cierta frontera de acuerdo al número de nodos que definen esta limitación |
| **Entradas** | |
| (N): La cantidad de zonas sobre las que se desea la consulta | |
| **Resultados** | |
| Nombre de la zona según los nodos que definen su frontera | |
| Número de nodos que definen la frontera | |
| **Estimación de complejidad temporal** | |
| Para este requerimiento se tiene que hacer una búsqueda por lo que se espera una complejidad del orden de O (n) y para el mejor caso O(n log n). | |

|  |  |
| --- | --- |
| **Nombre** | 4C. Gráfica ASCII - Porcentaje de datos faltantes para el primer semestre 2018 |
| **Resumen** | Este requerimiento supone la creación de una tabla ASCII que permita mostrar para cada zona de origen el porcentaje de datos faltante, es decir, el porcentaje de viajes que no se llevaron entre una zona origen y una zona destino para una hora y un trimestre dado |
| **Entradas** | |
| Hora en la que se dio un viaje | |
| Trimestre para el cual se desea hacer la consulta | |
| **Resultados** | |
| El resultado de este requerimiento se muestra en un formato especificado  1 | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  2 | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  3 | \*\*\*\*\*\*\*\*\*\*\*\*\*  Donde cada ‘\*’ representa un 2% | |
| **Estimación de complejidad temporal** | |
| Para este requerimiento se tiene que hacer una búsqueda por lo que se espera una complejidad del orden de O (n) y para el mejor caso O(n log n). | |