Trabajo Final de Estructuras de Datos – FAI - UNCOMA
Cursado 2022

Sistema de Trenes

Se desea desarrollar un sistema para brindar información a los usuarios de la red de trenes del país, llamado **TrenesSA**.

El principal objetivo del sistema será permitir consultas sobre qué camino tomar para llegar de una estación A a una estación B y también poder obtener información sobre las estaciones, trenes y líneas del servicio en particular.

La red de trenes estará conformada por un conjunto de estaciones unidas entre sí por rieles que permiten la circulación en ambos sentidos. Cada estación recibe un nombre único dentro del sistema (por ejemplo, "Retiro") y se desea almacenar la siguiente información de cada una: su domicilio (calle, número, ciudad y código postal), cantidad de vías y cantidad de plataformas.

La información de las estaciones se almacenará en un TDA Diccionario/Tabla de Búsqueda implementado con un Árbol AVL. En el caso de las estaciones la clave será el nombre de las mismas. El TDA AVL almacenará claves de tipo Comparable y dato de tipo Object.

Además se utilizará un grafo etiquetado para almacenar el mapa de las vías, que guarda las conexiones (rieles) entre las estaciones y la distancia entre ambas estaciones medida en kilómetros. El TDA grafo deberá mantener vértices de tipo Object y etiquetas de tipo numérico.

Por otro lado se utilizará un Mapeo (usando HashMap de Java) para almacenar las líneas de trenes y las estaciones que forman dicha línea. Tener en cuenta que una estación podría estar compartida por distintas líneas.

También se desea mantener información de qué trenes están trabajando en cada línea, por lo que en otro diccionario implementado con Árbol AVL, se almacenará para cada tren un identificador numérico único, el tipo de propulsión (electricidad, diesel, fuel oil, gasolina, híbrido), cantidad de vagones para pasajeros, cantidad de vagones para carga y la línea en la que está siendo utilizado (si el tren no está destinado a ninguna línea se considerará libre).

Se pide el desarrollo de una clase **TrenesSA** con un menú de opciones para realizar las siguientes tareas:

1. *Carga inicial del sistema*: ingresar al sistema un lote fijo de estaciones, trenes, líneas.
2. *ABM*[1] *de trenes*
3. *ABM de estaciones*
4. *ABM de líneas*
5. *ABM de la red de rieles*
6. *Consulta sobre trenes*:
   - Dado un código de tren mostrar toda la información del mismo
   - Dado un código de tren, verificar si está destinado a alguna línea y mostrar las ciudades que visitaría
7. *Consultas sobre estaciones*:

[1] ABM significa ALTAS-BAJAS-MODIFICACIONES. Para modificaciones considere sólo los datos mutables: no permita modificar las claves. Por ejemplo no debería cambiarse el nombre de una estación, línea o tren por ser claves del AVL, pero podría cambiarse la cantidad de vías o vagones.

- Dado un nombre de estación, mostrar toda su información
- Dada una cadena, devolver todas las estaciones cuyo nombre comienza con dicha subcadena (por ejemplo si la cadena es “Villa” debería considerar listar el rango desde “Villa” hasta “VillaZZZZ”)

8. *Consultas sobre viajes*: Dada una estación A y una estación B:
   - Obtener el camino que llegue de A a B que pase por menos estaciones
   - Obtener el camino que llegue de A a B de menor distancia en kilómetros
   - (*) Obtener todos los caminos posibles para llegar de A a B sin pasar por una estación C dada
   - (*) Verificar si es posible llegar de A a B recorriendo como máximo una cantidad X de kilómetros
9. *Mostrar sistema*: es una operación que permite ver todas las estructuras utilizadas con su contenido (grafo, árboles AVL y Mapeo) para verificar que se encuentran bien cargadas. No deben ser listados, sino debe poder verse la estructura tal cual está cargada. Por ejemplo, en el caso del árbol AVL debe ser visible la altura de cada nodo y quienes son los nodos hijo izquierdo y derecho de cada nodo.

Requisitos:

- El programa debe permitir la ejecución por separado de cada una de las operaciones especificadas.
- El programa **debe ser eficiente**: Debe recorrer las estructuras sólo lo necesario y haciendo buen uso de la memoria.
- Recordar **modularizar apropiadamente** apuntando a un código reusable y fácilmente mantenible.
- Las estructuras deben estar implementadas de forma genérica para elementos de tipo Object o Comparable de Java, según la necesidad del TDA.
- La carga inicial del sistema debe hacerse a partir de un archivo de texto con formato preestablecido y desde las opciones del menú se deben poder cargar datos adicionales. La carga inicial debe realizar un control de la consistencia de los datos (todas líneas tengan estaciones, el mapa de rieles tiene la información de las distancias, etc)
- Utilizar un *archivo de log* (archivo de texto) para guardar la siguiente información: estado del sistema (contenido de todas las estructuras) al terminar la carga inicial, a continuación anotar qué operaciones de ABM se realizan a lo largo de la ejecución (Ej: “Se creó la estación X”, “Se borró el tramo de riel entre las estaciones X e Y”, etc), y estado del sistema al momento de terminar de ejecutarse.

Condiciones y fechas de entrega:

- El trabajo debe realizarse de manera **individual** y debe presentarse **personalmente** a los docentes antes de subirlo a PEDCO en el curso “EDAT (preparando final)”.
- Al momento de la defensa, se deberá presentar un dibujo (en papel o digital) del mapa cargado (grafo) y de los diccionarios de estaciones y de trenes (AVL)
- Los **estudiantes que promocionan** la materia tendrán tiempo para entregarlo hasta el **viernes 5 de agosto de 2022** y no necesitan realizar los módulos marcados con (*)
- Los estudiantes que no promocionan podrán entregarlo en cualquier momento, pero como mínimo deberán hacerlo 2 semanas antes de presentarse a rendir el final regular.

Se puede tomar el siguiente link como ejemplo para cargar la red ferroviaria argentina, completando con estaciones que no existen en la actualidad, pero que serían muy útiles si existieran:

https://www.argentina.gob.ar/transporte/trenes

PARA INVESTIGAR EL USO DE JAVA

- Clase StringTokenizer (para fraccionar un String con un caracter separador)
- Para abrir archivo de texto para lectura: FileReader, BufferedReader o similares
- Para escribir en archivo de texto: FileWriter o similares

FORMATO DE ARCHIVO DE TEXTO (EJEMPLOS)

Estación: nombre de la estación, calle, número, ciudad, código postal, cantidad vías, cantidad plataformas

- E;Retiro; Av. Dr. José María Ramos Mejía; 1430; CABA; C1104; 9; 6
- E;Córdoba;Boulevard Juan Domingo Perón 101;Córdoba;5000;6;4
- E;Mar del Plata;San Juan 1642;Mar del Plata;7600;5;4

Línea: nombre de la línea y nombre de las estaciones por las que pasa

- L;Mitre;Retiro;Campana;Zárate;Baradero…
- L;Roca;Plaza Constitución;Haedo;Cañuelas…

Mapa de Rieles: nombres de las estaciones que se conectan y la distancia en kilómetros entre ellas

- R;Retiro;Campana;80
- R;Zárate;Baradero;74
- R;Zárate;Campana;80
- R;Córdoba;Rosario;400
- R;Plaza Constitución;Haedo;23
- R;Haedo;Cañuelas;65

Trenes: código único numérico, tipo de propulsión, cantidad de vagones para pasajeros, cantidad de vagones para carga y línea en la que está siendo utilizado (si el tren no está destinado a ninguna línea se considerará libre o no-asignado)

- T;234;diesel;5;6;Mitre
- T;6789;eléctrico;10;1;Roca
- T;893;gasolina;0;11;no-asignado

PARA LA ENTREGA LA CARGA INICIAL DEBE CONTAR **AL MENOS** CON 20 ELEMENTOS DE TIPO ESTACION, 20 TRENES Y 30 RIELES EN EL MAPA.

ASEGURAR QUE EN EL SET DE CARGA INICIAL LOS ELEMENTOS SE LISTEN EN FORMA DESORDENADA PARA QUE SE PRODUZCAN **TODAS LAS ROTACIONES POSIBLES** EN CADA AVL.