

Algoritmos y Programación II

1C 2020

Trabajo Práctico 5



El trabajo práctico 5 (último trabajo) consiste en realizar una aplicación para el manejo de aeropuertos y vuelos.

Aeropuertos

La información de los aeropuertos estará dada en un archivo de texto *aeropuertos.txt* que deberá leerse al iniciar la aplicación. Esta lectura debe generar un diccionario implementado en un árbol binario de búsqueda (ABB).

En cada línea del archivo estará la información de un aeropuerto, separada por espacios:

```
codigo_IATA nombre ciudad pais superficie #cantidad_terminales #destinos_nacionales #destinos_internacionales
```

El código IATA es un código de tres letras que identifica a cada aeropuerto. Por ejemplo, EZE es el aeropuerto Ministro Pistarini de Ezeiza, COR es el aeropuerto Pajas Blancas de Córdoba.

La superficie es un número flotante que indica los km². Los últimos tres datos son números enteros con las cantidades correspondientes. Ejemplo:

```
EZE Ministro-Pistarini Ezeiza Argentina 34.75 4 12 46
LAX Aeropuerto-Internacional-de-los-Angeles Los-Angeles EEUU 14 9 87 69
...
```

En la primera línea tenemos al aeropuerto Ministro-Pistarini de Ezeiza (EZE), con 34,75 km² de superficie, 4 terminales, 12 destinos nacionales y 46 internacionales.

Los nodos del árbol deben guardar la clave (el código IATA) y un puntero adonde están los datos del aeropuerto.

Luego de cargar el árbol desde el archivo, debe mostrar un menú en el cual se pueda

- consultar por un aeropuerto en particular
- dar de alta un nuevo aeropuerto
- dar de baja a alguno
- mostrar todos los aeropuertos (recorrido in orden)

- mostrar los aeropuertos conservando el formato del ABB (recorrido en ancho). Es decir, en la primera línea mostrará un aeropuerto (la raíz del ABB), en la segunda, dos: los hijos de la raíz, en la tercera, cuatro, etc. (esto sería en el caso en que el árbol esté balanceado, podría no estarlo). En este recorrido solo deben imprimirse los códigos IATA (la clave del diccionario).

Consideraciones

- El archivo está bien formado.
- La implementación puede ser recursiva o iterativa.

Vuelos

A partir de la lectura de un archivo de texto *vuelos.txt*, que deberá hacerse inmediatamente después de leer el archivo aeropuertos.txt, se generará un grafo pesado dirigido.

En cada línea estará la información de cada vuelo, separada por espacios:

```
codigo_IATA_partida código_IATA_destino costo_vuelo horas_vuelo
```

El costo del vuelo es un entero que representa el costo en pesos y las horas de vuelo será un flotante.

Ejemplo:

```
EZE MIA 122483 9.2
EZE FCO 222205 13
FCO TXL 4736 2.25
...
```

La primera línea indica que hay un vuelo directo de Ezeiza a Miami que dura 9,2 hs, cuyo precio es \$122.483.

La segunda, de Ezeiza a Fiumicino, Roma, dura 13 horas y sale \$222.205.

La tercera, de Fiumicino a Otto-Lilienthal, Berlín, dura 2 horas y cuarto y su precio es de \$4.736.

La aplicación debe pedir al usuario el ingreso de los códigos IATA de partida y de destino, y debe buscar la combinación de vuelos más económica o de menor tiempo. El usuario debe poder elegir si el camino mínimo está representado por el costo del vuelo o por la duración.

Se debe imprimir el detalle del vuelo, el costo de cada tramo (o el tiempo de vuelo) y el total del viaje.

Si hubiera más de una combinación óptima, deberá listarlas todas.

Puede suceder que no haya ninguna combinación, en ese caso se informa que no hay conexiones posibles.

Consideraciones

- El archivo está bien formado.
- La implementación puede ser recursiva o iterativa.

A tener en cuenta

- Separación de archivos (h y cpp)
- Funcionalidad
- Interfaz con el usuario
- Nombres de métodos y atributos
- Pre y post condiciones
- Modularización
- Utilización de memoria dinámica
- Buenas prácticas de programación
- Diseño UML

Nota: la aplicación debe estar completamente orientada a objetos, a excepción del main, en donde solo puede tener la creación de unos pocos objetos y el llamado a algunos de sus métodos.

Entrega

Semanalmente (martes o jueves), cada equipo deberá realizar una exposición del avance del proyecto con alguno de los docentes del curso. Por ejemplo, para la primera semana sería ideal contar con un diseño de clases, el o la responsable de dicha clase, la división de tareas, un cronograma con los hitos principales y, de ser posible, el armado del diccionario. Luego, cada docente irá indicando los objetivos de la siguiente semana.

La entrega final del trabajo será, o bien el martes 11 o el jueves 13 de agosto, en horario de clases, momento en que haremos la defensa del mismo. Por defensa se entiende la muestra del trabajo junto con una explicación por parte de cada uno de los integrantes.

El puntaje máximo de este trabajo otorga 40 puntos. El puntaje puede ser diferente para cada integrante del grupo dependiendo de la dedicación realizada.