



SIMULACIÓN DEL CONTROL DE ADUANAS VERSIÓN II

En esta práctica se pretende simular el funcionamiento del control de aduanas en un aeropuerto para atender a pasajeros que llegan a diferentes horas del día. Suponemos que cada pasajero que llega al aeropuerto tiene:

- Una hora de llegada que es indicada en minutos y que se cuentan desde las 00:00 de la noche. Por ejemplo, tiempo de llegada igual a 77 quiere decir que el pasajero llegará a las 1:17.
- Duración del control de aduanas, también en minutos. Si el pasajero anterior tiene una duración de 20 minutos, habrá sido atendido a las 1:37.

Como resultado final queremos calcular el tiempo medio que los pasajeros **esperan** para ser **atendidos** dentro del aeropuerto. Los pasajeros se atienden en función de su hora de llegada.

Partiendo de los pasajeros del ANEXO I, se debe crear un **árbol binario** de búsqueda (ABB), implementado dinámicamente, que almacenará dichos pasajeros ordenados por su id (no es necesario usar ningún pasajero ficticio). Una vez creado el árbol, se debe recorrer en inorden almacenando los pasajeros en una **pila** implementada dinámicamente.

Partiendo de la pila de pasajeros anterior, se pide simular una **lista** implementada dinámicamente, y que el sistema esté gestionado en función de los dos siguientes eventos:

- Llegada de un pasajero nuevo a aeropuerto.
- Salida de un pasajero del aeropuerto (cuando termina su tiempo de atención).

Los pasajeros que llegan antes serán colocados en la parte alta de la lista para ser los primeros en ser atendidos.

Los pasajeros se van atendiendo en único box. Cuando se produce un evento de llegada, si el box está ocupado, el pasajero se inserta en la lista en función de su hora de llegada. Cuando un pasajero sale del box, entra en dicho box el primer pasajero de la lista (que corresponde al de menor hora de llegada).

Los pasajeros que van saliendo del box se almacenan finalmente en una **cola** implementada dinámicamente que se deberá recorrer para calcular el tiempo medio de espera de los pasajeros.

Para cada pasajero se almacena: *id de pasajero*, *hora de llegada* y *la duración de la atención* (ambos en minutos).

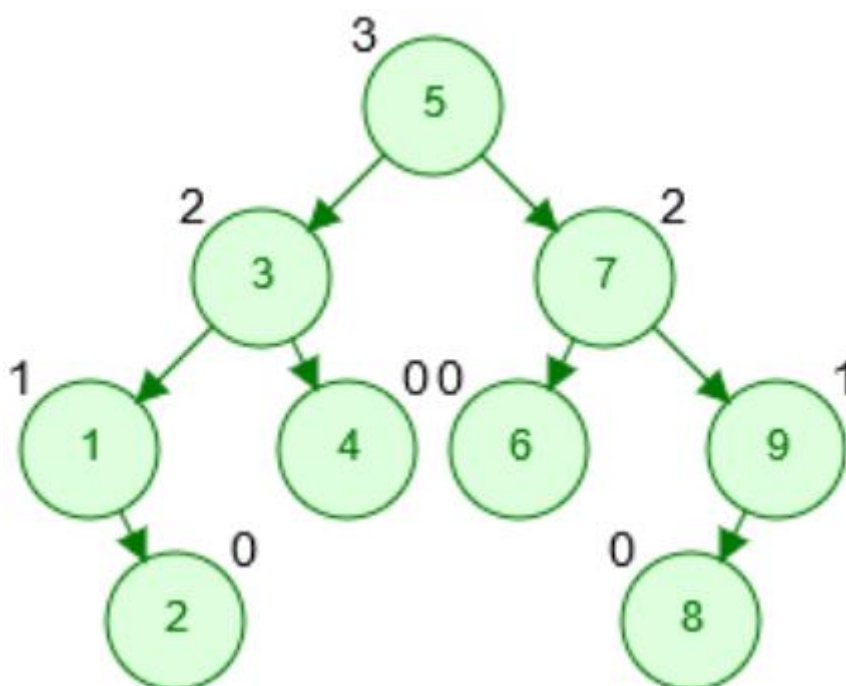
Se pide implementar una clase Box, una clase Pasajero, una clase Pila, una clase Cola, una clase Lista, una clase Árbol y una clase Aeropuerto, así como un programa principal que invoque una función "llegada" y una función "salida" que simule los dos tipos de eventos. El programa principal debe simular el funcionamiento del control de aduanas: se le debe dar como entrada la llegada de diferentes pasajeros con su id, su minuto de llegada y la duración de su atención en el box y debe producir como salida el tiempo medio que estas personas han esperado dentro del aeropuerto.

ANEXO I: Ejemplo de la simulación de pasajeros

Supongamos que tenemos los siguientes pasajeros:

PASAJERO	id	HORA DE LLEGADA	DURACIÓN
1	5	15	5
2	3	22	6
3	7	9	7
4	6	14	3
5	4	18	4
6	1	27	17
7	9	0	6
8	2	26	3
9	8	4	7

Árbol creado en función del id de cada pasajero:



Al acabar de atender a todos los pasajeros del ejemplo, el tiempo que ha esperado cada uno en el aeropuerto es el siguiente:

PASAJERO	1	2	3	4	5	6	7	8	9
TIEMPO	16	14	12	10	8	6	4	2	0

Con lo que el tiempo medio de atención es de $72 / 9 = 8.0$ minutos.



NORMAS PARA LA REALIZACIÓN DE LA PRÁCTICA

1. Las prácticas se realizarán preferentemente en grupos de dos alumnos que deberán ser indicados por correo electrónico a hamid.tayebi@uah.es antes del 15 de mayo de 2020.
2. Debe entregarse un fichero comprimido incluyendo el **proyecto Codelite** y el documento descrito en el punto 6. Se subirá un fichero por grupo a la plataforma **antes de las 23:00 del 15 de junio de 2020**. El nombre del fichero será el Apellido1Apellido2Nombre de uno de los miembros del grupo.
3. La asistencia y realización de la defensa, es individual y obligatoria para todos los alumnos del grupo, siendo necesario traer la memoria en papel para la realización de la misma. En el caso de no asistir, la práctica se calificará con un suspenso (0).
4. En la **defensa** de la práctica **se verificará la autoría de la práctica entregada y será calificada con APTO/NO APTO, siendo necesaria la calificación de APTO para poder ser evaluado de la práctica**. La defensa se realizará a la hora y en el lugar indicado por el profesor.
5. La entrega de prácticas copiadas supondrá el suspenso de la asignatura en esta convocatoria para todos los alumnos implicados.
6. La documentación que se entregará impresa, junto con la defensa, deberá tener al menos los siguientes apartados:
 - a. Nombre y DNI de los alumnos del grupo.
 - b. Detalles y justificación de la implementación:
 - b.1 Especificación concreta de la interfaz de los TAD's implementados:
 - b.1.1 TAD's creados.
 - b.1.2 Definición de las operaciones del TAD (Nombre, argumentos y retorno).
 - b.2 Solución adoptada: descripción de las dificultades encontradas.
 - b.3 Diseño de la relación entre las clases de los TAD implementados.
 - b.3.1 Diagrama UML.
 - b.3.2 Explicación de los métodos más destacados.
 - b.4 Explicación del comportamiento del programa.
 - b.5 Bibliografía.