ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

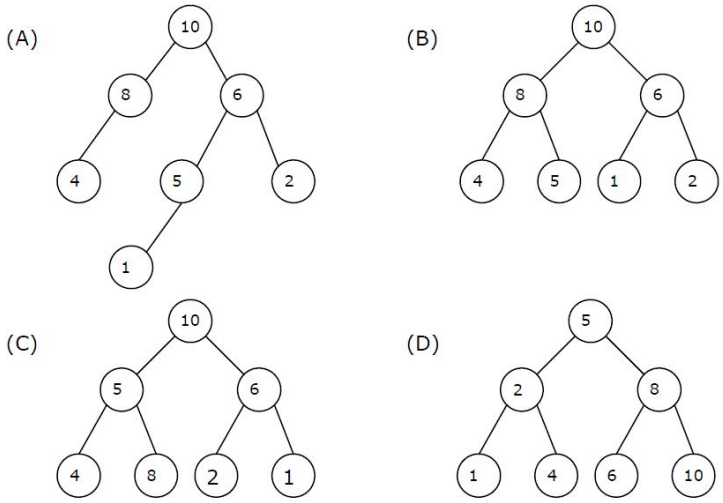
FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN
ESTRUCTURAS DE DATOS
EXAMEN DE MEJORAMIENTO - II TÉRMINO 2017

Nombre: _____ ***Matrícula:*** _____

TEMA 1. Elija la opción correcta y justifique su respuesta.(10 PUNTOS)

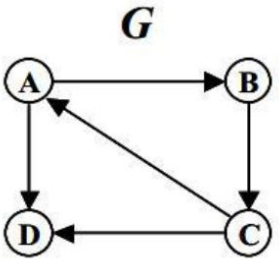
A. Suponga que los siguientes elementos son ingresados a un árbol binario de búsqueda 7, 5, 1, 8, 3, 6, 0, 9, 4, 2. Cuál es el resultado de hacer un recorrido en orden del árbol.
   a. 7 5 1 0 3 2 4 6 8 9
   b. 0 2 4 3 1 6 5 9 8 7
   c. 0 1 2 3 4 5 6 7 8 9
   d. 9 8 6 4 2 3 0 1 5 7

B. Cuál de los siguientes árboles es un max heap.



C. Considere una tabla de tamaño 7 y se desea insertar los siguientes elementos 2341, 4234, 2839, 430, 22, 397, 3920. Cuál es el resultado de la tabla luego de aplicar una función hash h(x) = x%7 y una función rehash lineal.

D. Cuáles son las componentes conexas del siguiente grafo.



E. Cuántos bits son necesario para codificar con huffman la palabra “mississippi”.

TEMA 2. (10 PUNTOS)
En el TDA Lista Simplemente enlazada, Implementar el método removerUltimaOcurrencia. El método recibe un elemento y remueve la última ocurrencia de ese elemento en la lista. El algoritmo debe tener un tiempo de ejecución O(n).

TEMA 3. (25 PUNTOS)
League of Legends (LoL) es un videojuego de campo de batalla multijugador en línea. En el juego existen varios tipos de batallas, entre los que se tiene el enfrentamiento en equipos (Matchmaking). Este tipo de batalla enfrenta a 2 equipos de 5 jugadores cada uno, en donde cada jugador es representado por un personaje. Cada personaje tiene un grupo de características como tipo, vida, nivel, ataque, etc.

Suponiendo que el servidor del juego tiene una lista de todos los jugadores (personajes) que se desean unir a un nuevo Matchmaking, la cual está organizada según el orden en que cada jugador realizó la solicitud de batalla:

Implemente la creación de un Matchmaking en donde los jugadores no pueden diferir en más de n niveles entre sí. El método toma como parámetro la lista de jugadores en espera del servidor y retorna una batalla tomando como base el primer jugador que realizó una solicitud de batalla. Si no existen suficientes jugadores que difieren en n niveles, se deben comenzar a agregar jugadores que difieren en n+1 niveles, n+2 niveles, n+3 niveles, etc hasta completar todos los jugadores para la batalla. Una batalla consiste de dos listas de 5 jugadores cada una. Además, el método matchmaking actualiza la lista del servidor.

public static Batalla matchmaking(List<Jugador> jugadores, int n)

TEMA 4. (25 PUNTOS)

En videojuegos se conoce como NPC - “non playable character” a un personaje no jugador, los cuales son personajes controlados por el computador. Uno de los métodos para determinar la operación de los NPC son los árboles de comportamiento, los cuales contienen las tareas a ejecutar en sus hojas. Los demás nodos de un árbol de comportamiento pueden cumplir distintas funciones, dos de estas funciones son selectores y secuenciadores. Los nodos selectores ejecutan un hijo según las condiciones de una variable de ambiente (hijo izquierdo si variable es true, hijo derecho si variable es false). Los nodos secuenciadores ejecutan los hijos uno por uno (primero el izquierdo y luego el derecho).

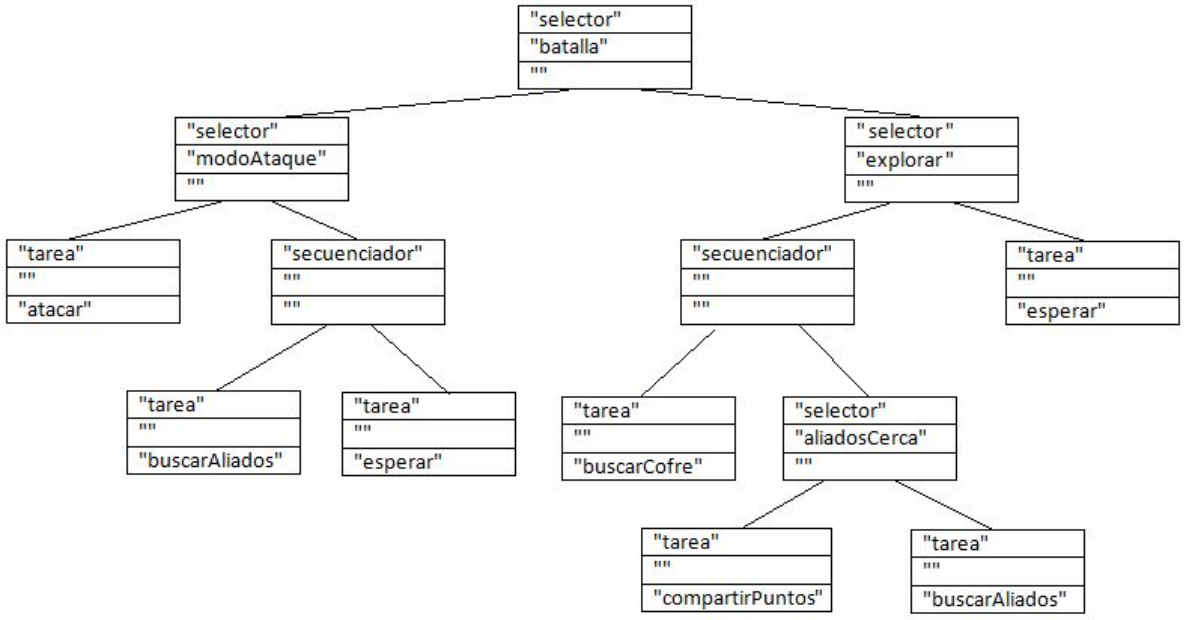
```
public class Nodo {
        String tipo;
        String variableID;
        String tareaID;
        Nodo izq, der;
}
```

```
public class Arbol{
        Nodo raiz;
}
```

El atributo tipo puede ser "tarea", "selector" o "secuenciador"
El atributo variableID contiene el ID de la variable a utilizar si el nodo es de tipo "selector"
El atributo tareaID contiene el ID de la tarea a ejecutar si el nodo es de tipo "tarea"

Escribir la secuencia de ID de las tareas que ejecutará un NPC siguiendo un árbol de comportamiento y el mapa de variables de ambiente.

public List<String> ejecutarTareas(AB arbol, HashMap <String, boolean> variablesAmbiente)



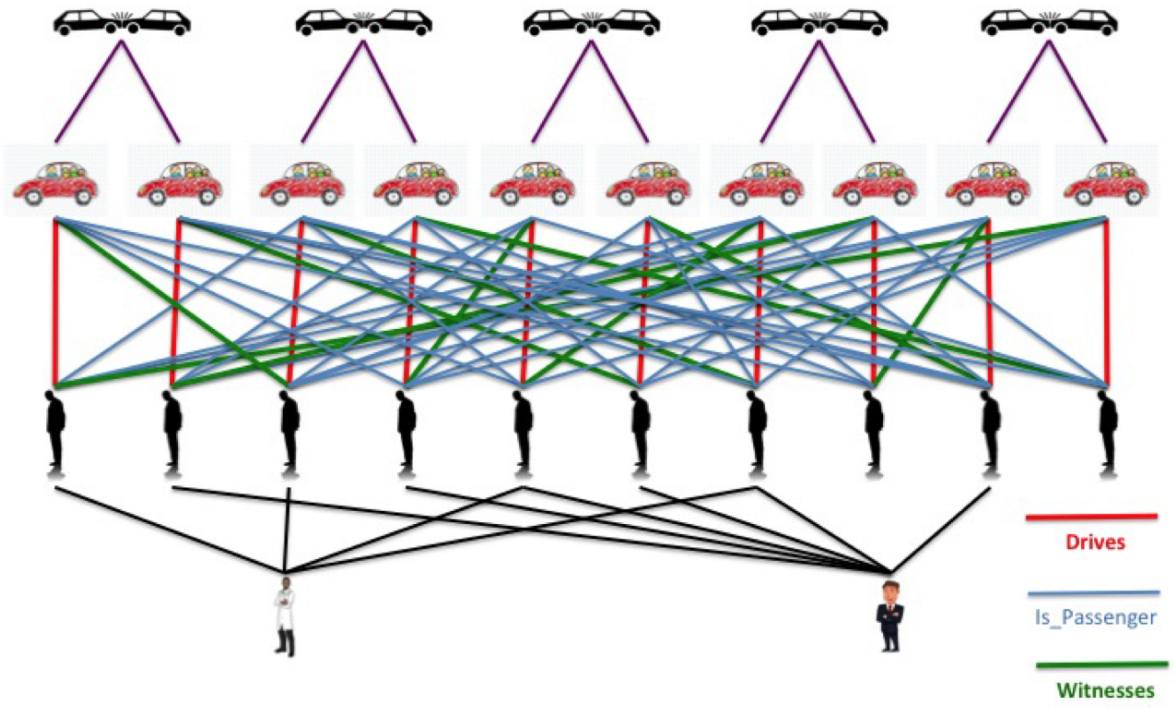
ambiente1 = {	ambiente2 = {	ambiente3 = {
"batalla": true,	"batalla": true,	"batalla": false,
"modoAtaque": false,	"modoAtaque": true,	"modoAtaque": true,
"explorar": false,	"explorar": false,	"explorar": true,
"aliadosCerca": true	"aliadosCerca": true	"aliadosCerca": true
}	}	}
retorna:		
["buscarAliados", "esperar"]	["atacar"]	["buscarCofre", "compartirPuntos"]

TEMA 5. (30 PUNTOS)

Las aseguradoras de carros quieren determinar si han sido víctimas de fraudes en el cobro de seguros por accidentes, ya que tienen una sospecha de que los abogados y los doctores se han puesto de acuerdo con una red de personas para estar involucradas constantemente en accidentes de carros. La aseguradora ha construido un grafo no dirigido implementado con una lista de adyacencia, cuyos vértices pueden ser objetos de las clases: Persona, Carro, Accidente. A usted se le solicita implementar las siguientes funciones:

A. El método estático **númeroAccidentes** que recibe el Grafo y una Persona (abogado, doctor, o conductor). La función determina en cuantos accidentes ha estado involucrado esa persona.
B. El método estático **conductoresSospechozos** que recibe el Grafo y retorna una Lista de Personas conductores que tienen más de 2 carros y han estado involucrado en más de 3 accidentes.

Nota.- Puede utilizar el operador instanceof para determinar la clase del objeto en el vértice.


Drives
Is_Passenger
Witnesses