

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN  
ESTRUCTURAS DE DATOS  
EVALUACIÓN DE MEJORAMIENTO - I TÉRMINO 2017

Nombre: \_\_\_\_\_ Matrícula: \_\_\_\_\_

TEMA 1. (10 PUNTOS)

Elija la opción correcta y **justifique** su respuesta a las siguientes preguntas:

1. Considerando el siguiente método:

Cual operación producirá la mayor cantidad de llamadas recursivas?

- a. recursivo(-1023)
- b. recursivo(0)
- c. recursivo(100)
- d. recursivo(1023)

```
void recursivo(int n){  
    if (n < 0)  
        recursivo(-n);  
    else  
        if (n < 10)  
            System.out.println(n);  
        else  
            recursivo(n/10);  
}
```

2. Recursión utiliza más memoria que iteración debido a que:

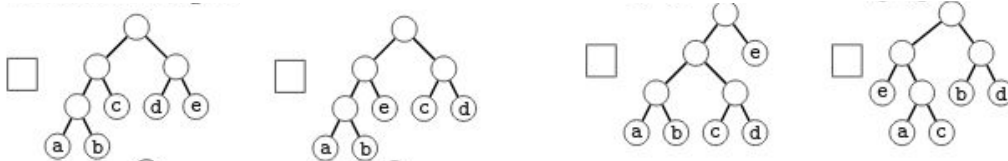
- a. Usa pilas en lugar de colas.
- b. Cada llamada recursiva tiene que ser almacenada.
- c. Ambas respuestas (a y b) son correctas.
- d. Ninguna de las anteriores.

3. Que valor imprime en pantalla el siguiente código:

- a. A
- b. B
- c. C
- d. Ninguna de los anteriores.

```
String[] states = {"A", "B", "C"};  
ArrayList list = new ArrayList ();  
ArrayList(Arrays.asList(states));  
ListIterator it = list.listIterator();  
System.out.println(it.previous());
```

4. Un mensaje con las letras a,b,c,d,e es comprimido usando código de Huffman.Las frecuencias de a,b,c, y d es 14 cada uno, mientras que la de e es 28. Cual de los siguientes árboles NO podría resultar del algoritmo de Huffman?



5. A un árbol AVL vacío se se ingresan los siguientes valores: 1,2,3,8,6. Cuántas rotaciones se realizan?

- a. Ninguna.
- b. Una rotación simple.
- c. Una rotación doble.
- d. Una simple y una doble.

## TEMA 2. (10 PUNTOS)

Considere las siguientes estructuras:

|                          |                           |                              |
|--------------------------|---------------------------|------------------------------|
| Lista enlazada           | Árbol                     | Grafo Dirigido sin ciclos    |
| Lista basada en arreglos | Árbol Binario de Búsqueda | Grafo Dirigido con ciclos    |
| Mapa                     | Árbol AVL                 | Grafo No Dirigido con ciclos |
| Conjunto                 | Heap                      | Grafo No Dirigido sin ciclos |
| Cola                     | Cola de Prioridad         | Pila                         |

Seleccione y justifique la o la combinación de estructura de datos más apropiada para cada uno de los siguientes problemas:

1. Las notificaciones de las diferentes aplicaciones (facebook, twitter, whatsapp, etc) que llegan a un smartphone. El usuario puede descartar cualquiera de las notificaciones si ya las ha leído.
2. La malla curricular de una carrera de una universidad, donde cada materia tiene pre-requisitos, co-requisitos, créditos teóricos, créditos prácticos.
3. Facebook sugiere nuevas amistades a partir de los amigos de los contactos que uno tiene agregado.
4. En una competición de natación se han tomado los tiempos de todos los nadadores que han concursado en los diferentes estilos (libre, espalda, pecho, mariposa). Se desea premiar a los tres mejores tiempos de cada estilo.
5. La opción de autocompletar a medida que vamos escribiendo los nombres de las funciones cuando programamos en java.

## TEMA 3. (20 PUNTOS)

Una empresa tiene instalada una central telefónica que lleva el registro de las llamadas telefónicas en una pila de strings, cada string de la pila representa un registro de llamada desde una extensión origen a varias extensiones destino con un tiempo de duración.

**extOrigen|extDestino-Tiempo|extDestino-Tiempo.....**

"300|700-15|800-5"

"600|800-50"

"300|500-80|700-20|800-2"

A usted se le solicita crear una función estadística que recibe una pila de strings, y retorna un mapa cuyas claves son las extensiones orígenes y el valor es una lista de extensiones destino ordenadas ascendentemente por su tiempo de duración total. (asuma que ya existe el TDA Destino)

```
public class Destino {  
    private int extDestino, duracion;  
    ...}
```

```
public static HashMap<Integer,LinkedList<Destino>> estadistica(Stack<String> registros)
```

Ejemplo:

```
{300 : [800-7, 700-35, 500-80], 600 : [800-50]}
```

#### TEMA 4. (20 PUNTOS)

El despegue de aeronaves en un aeropuerto se realiza siguiendo el orden establecido por 3 prioridades según la distancia del destino:

Prioridad 1: Destinos de menos de 500 km (máxima prioridad)

Prioridad 2: Entre 500 y 1000 km

Prioridad 3: Distancias mayores a 1000 km

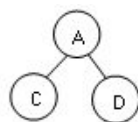
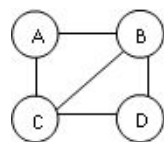
Cada aeronave que desea despegar es almacenada en la lista de espera según su prioridad y empieza a contar el tiempo de espera. Cada 5 minutos la aeronave de mayor prioridad y que ha esperado por mayor tiempo tiene autorización de despegue. El piloto de la aeronave a despegar puede activar un aviso si tiene un problema y no puede despegar, por lo que pasa la final de la lista de espera correspondiente y se da la orden de despegue a la siguiente aeronave. Puede darse la circunstancia de que una aeronave lleve más de 20 minutos esperando; en ese caso pasará a formar parte de la siguiente lista de espera y su tiempo se inicializará en 0.

Escriba un programa para simular este sistema (asuma que la llamada a cada iteración se realiza cada 5 minutos).

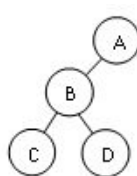
#### TEMA 5. (25 PUNTOS)

Dado un grafo conexo G y un árbol binario A, se dice que A es un árbol de expansión de G si:

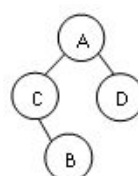
- Los nodos de A no están repetidos.
- El conjunto de nodos de A es igual al conjunto de vértices de G.
- Los arcos de A son arcos en el grafo y por lo tanto permiten llegar desde la raíz a cualquier otro nodo.



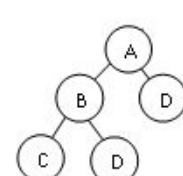
False



True



False



False

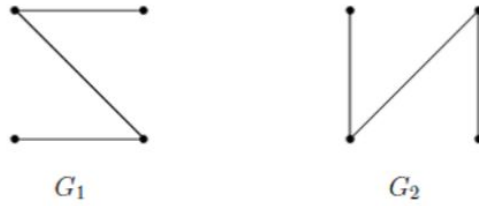
Se le solicita que en el TDA Grafo, implemente el método **esArbolExpansion**, que recibe un árbol binario y retorna verdadero si el parámetro recibido es un árbol de expansión del grafo. Para esto, completar los siguientes pasos:

1. Escribir un método que retorna el conjunto de nodos del árbol. Si existen nodos repetidos se retorna un conjunto vacío. (5 puntos)
2. Escribir un método que retorna verdadero si el conjunto de nodos de A es igual al conjunto de vértices de G. (5 puntos)
3. Escribir un método que valida si los arcos de A son arcos en el grafo G. (10 puntos)
4. Implementar el método **esArbolExpansion (AB arbol)**. (5 puntos)

## TEMA 6. (15 PUNTOS)

Implementar el método **grafoComplementario** en el TDA Grafo con Lista y Matriz de adyacencia que retorna un nuevo grafo no dirigido que es el complemento del grafo original.

```
public Grafo<E> grafoComplementario()
```



Grafos complementarios



