

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

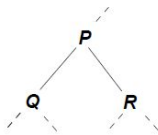
FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN  
 ESTRUCTURAS DE DATOS  
 SEGUNDA EVALUACIÓN - II TÉRMINO 2017

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

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

a. Un Heap es un árbol binario que satisface la condición de ser "parcialmente ordenado". Si P, Q, R son las etiquetas del nodo y de sus dos hijos, la condición de parcialmente ordenado se expresa como (Nota: Consideramos un heap ascendente):

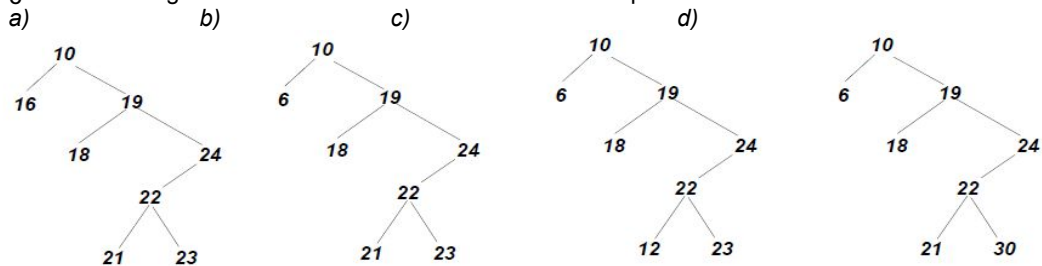
- a)  $Q + R < P$
- b)  $Q < P < R$
- c)  $P < (Q + R)$
- d)  $P < Q \ \&\& \ P < R$



b. El algoritmo de Huffman permite obtener códigos binarios para codificar mensajes utilizando árboles binarios. La longitud del código asignado a un carácter, en bits está dado por:

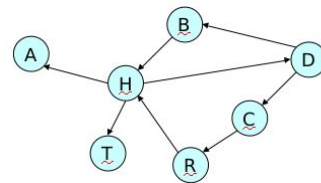
- a) la profundidad del nodo correspondiente en el árbol.
- b) la cantidad de letras en el alfabeto utilizado
- c) la etiqueta del nodo correspondiente en el árbol.
- d) el número de nodos en el subárbol del nodo correspondiente.

c. ¿Cuál de los siguientes árboles es un árbol binario de búsqueda?

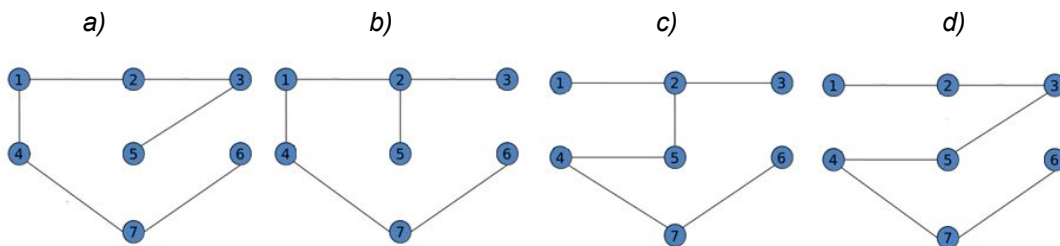
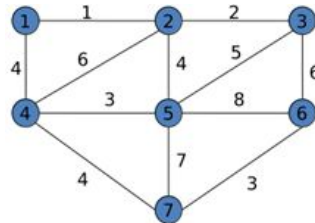


d. ¿Cuál de los siguientes recorridos NO es un posible recorrido en anchura del grafo mostrado?

- a) DCBHRTA
- b) HATDCBR
- c) CRHATDB
- d) DBHCRTA



e. ¿Cuál de los siguientes árboles es un posible árbol de expansión de coste mínimo del grafo mostrado?



## TEMA 2. (10 PUNTOS)

Considere las siguientes estructuras:

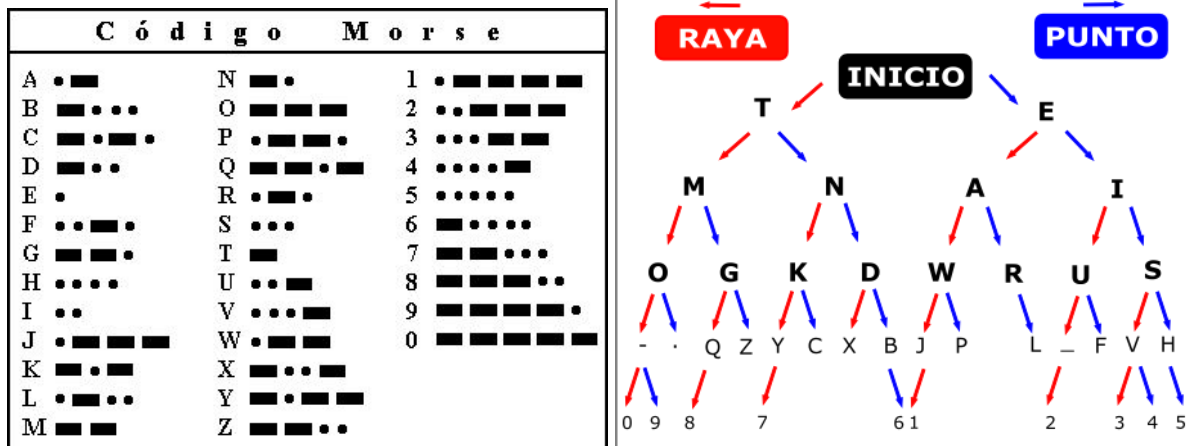
Árbol Multicaminos	Grafo Dirigido sin ciclos
Árbol Binario	Grafo Dirigido con ciclos
Árbol Binario de Búsqueda / AVL	Grafo No Dirigido con ciclos
Heap	Grafo No Dirigido sin ciclos

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

1. La funcionalidad de autocompletar nombres de comandos cuando se utiliza la consola. Por ejemplo: cuando el usuario escribe en la consola “cop”, luego presiona la tecla TAB, y se autocompleta al comando “copy”.
2. Una empresa de banano tiene muchas haciendas productoras de banano. La empresa desea conocer cuál es la hacienda que menos produce y la que más produce banano.
3. Un director técnico de fútbol requiere un sistema informático que le permita analizar la cantidad de buenos y malos pases que realiza un jugador del equipo.
4. Whatsapp desea implementar un módulo de sugerencia de contactos para los usuarios de este servicio. Este módulo recomendará números de teléfonos de usuarios que son amigos de los contactos que uno tiene en whatsapp.
5. El CNE desea conocer las 3 provincias que han registrado el mayor porcentaje de ausentismo (% de personas que no ejercieron el voto).

## TEMA 3. (25 PUNTOS)

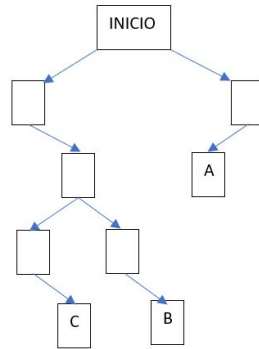
El código morse es un sistema de representación de letras, números y símbolos mediante señales emitidas intermitentemente. Estas señales se representan por una combinación de puntos y rayas que pueden ser representadas por un árbol binario.



- a) Se le solicita implementar la función crearArbolMorse. La función debe crear un árbol de morse a partir de un mapa que contiene todos los símbolos con sus respectivos códigos. Un ejemplo del mapa es el siguiente:

**mapa = {"A" : [".", "-"], "B" : ["-", ".", ".", "."], "C" : ["-", ".", "-", "."], ...}**

Una ilustración del árbol luego de procesar las tres primeras letras del mapa es el siguiente:



```
public static AB<String> crearArbolMorse(HashMap<String, List<String>> mapa)
```

b) Implementar la función `codificarMorse` que recibe una lista de códigos de morse (puntos y rayas) y el árbol de morse creado en el literal anterior. La función debe retornar la palabra que representan la secuencia de códigos de morse. Por ejemplo:

**codificarMorse(["-",".", ".", ".", ".", "-", "-", ".", "-","."], arbolMorse) retorna "BA"**

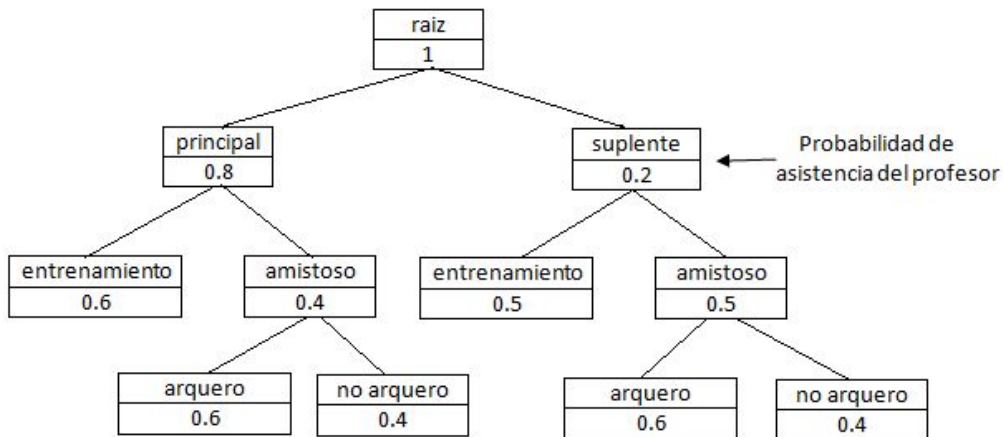
```
public static String codificarMorse(List<String> codigos, AB<String> arbol)
```

#### TEMA 4. (20 PUNTOS)

Iker Higueta es un estudiante politecnico que se encuentra cursando Futbol y al que no le agrada jugar de arquero. El sabe que cada vez que se juega un partido amistoso en clase, el 60% de las veces sus compañeros lo hacen tapar. La probabilidad de jugar un amistoso depende del profesor con el que entrenen ese día. Cuando entrenan con el profesor principal tienen amistosos el 40% de las veces, mientras que cuando entrenan con el profesor suplente juegan la mitad de las veces. Además, el profesor principal suele enviar al profesor suplente el 20% de las clases de forma aleatoria.

En base a los datos descritos, Iker ha creado el árbol de probabilidades mostrado en la figura siguiente. Utilizando el árbol de probabilidades se puede conocer que la probabilidad de que Iker sea arquero en cualquier día de clases es 25.2%, la cual está dada por la suma de todas las probabilidades de ser arquero (nodos arquero):

$$1 * 0.8 * 0.4 * 0.6 + 1 * 0.2 * 0.5 * 0.6 = 0.252$$



Con las siguientes definiciones del TDA Arbol y Nodo:

```
public class Nodo {
    String evento;
    double probabilidad;
    Nodo izq , der;
}

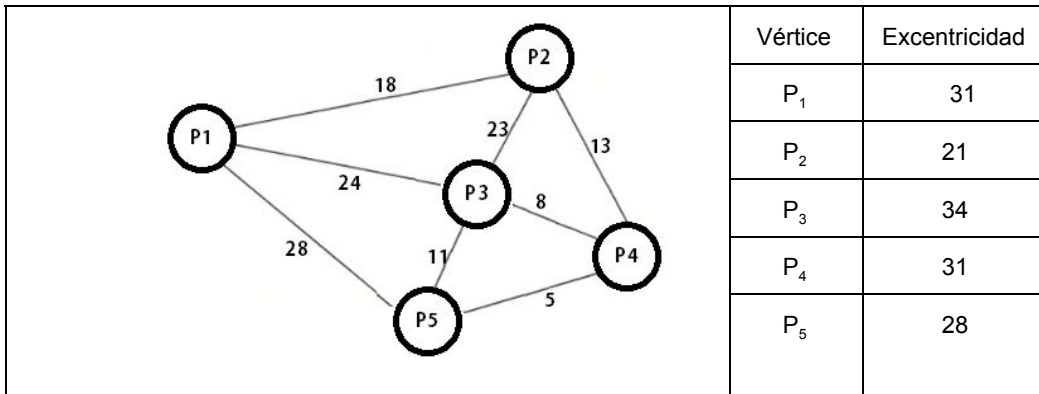
public class Arbol {
    Nodo raiz;
}
```

Implemente el método `estimarProbabilidad`, que determina la probabilidad de que un evento suceda. Por ejemplo, `estimarProbabilidad("arquero")` retorna 0.252.

## TEMA 5. (35 PUNTOS)

Se desea construir un hospital que dé cobertura a un grupo de poblados de similar número de habitantes de tal manera que todos los ciudadanos de los poblados lo puedan acceder, en promedio, lo más rápidamente posible. Por lo que, el lugar ideal para la construcción es el poblado que tenga la menor excentricidad dentro del grafo de poblaciones.

La excentricidad de un vértice  $v$  se define como la distancia de  $v$  al vértice más lejano, considerando solamente los caminos óptimos entre cada par de vértices. Por ejemplo, para el grafo de la figura, los caminos óptimos desde el poblado  $P_1$  a los demás poblados son:  $P_2$  - 18,  $P_3$  - 24,  $P_4$  - 31,  $P_5$  - 28. Por lo tanto, la excentricidad de  $P_1$  es 31 (el mayor de los caminos óptimos). Si se calcula la excentricidad para cada vértice del grafo se obtienen los siguientes valores:



El vértice con menor excentricidad es  $P_2$  con valor 21. Por lo tanto, si se construye el hospital en el poblado  $P_2$  se puede asegurar que todos los ciudadanos de los poblados lo puedan acceder viajando a lo mucho 21 kilómetros.

Implementar en el TDA Grafo:

- método estático: Grafo <Poblado> crearGrafo(List <Poblado> poblados), que retorna un grafo a partir de una lista de poblados. Cada TDA poblado contiene un id, un nombre y la función distanciaCarretera\* que retorna la distancia de la carretera que conecta directamente un poblado con otro, en caso de no existir carretera directa la función distanciaCarretera retorna -1. El Grafo se puede representar con Lista de adyacencia o Matriz de adyacencia, en ambos casos los poblados deben tener el mismo orden de la lista que se recibe como parámetro.

\*int distanciaCarretera(Poblado p) -> función existente en el TDA Poblado.

- List<Integer> caminosOptimos (Poblado p) retorna una lista de distancias de los caminos óptimos desde el poblado p a todos los poblados del Grafo (Se asume que el Grafo es conexo).
- int excentricidad (Poblado p) retorna el valor de la excentricidad del poblado p.
- String ubicacionIdeal() retorna el nombre del poblado en donde se debería ubicar al hospital, según las consideraciones antes descritas.
- Escribir un programa Main que a partir de una lista de ciudades imprima en pantalla:  
Cuál es la ciudad ideal para construir un hospital.  
Desde qué poblado se debe viajar la mayor distancia para llegar al hospital.



