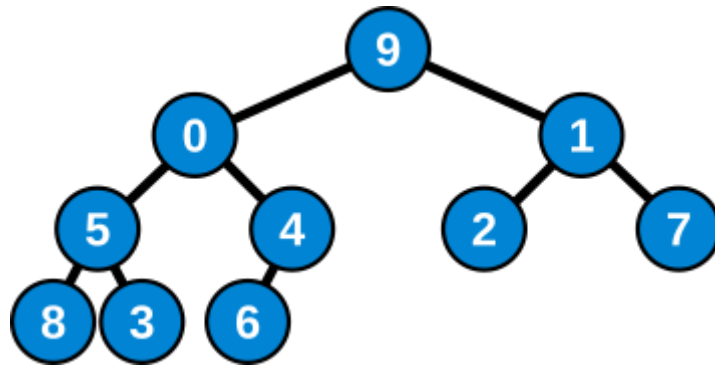


FINAL 25/03/21 - Algoritmos y Programación II

- 1) Indique si el siguiente árbol es un heap binario y si no lo es, muestre cómo transformarlo en uno. Muestre su representación vectorial y explique sus ventajas. Justifique.



- 2) Dado el vector de enteros `[1,7,4,3,9,8,23,5,4,0,11,24,5,6]` usar una combinación de **quicksort** y **mergesort** para ordenarlo (vaya alternando entre los dos algoritmos en cada paso, empezando por **mergesort**). Muestre el paso a paso, explique y justifique.

- 3) Dados los siguientes elementos:

```
Tabla = [L1,L2,L3];
```

```
L1 = [];
```

```
L2 = [<13;"Nico">, <19;"Tomas">];
```

Indique si la información es suficiente o no para definir una **Tabla de hash**. Si no lo es, explique qué información falta y agreguela. Caracterice la tabla. Justifique.

Agregue los siguientes pares a la tabla:

```
<1;"Mauricio">
```

```
<5;"Agustina">
```

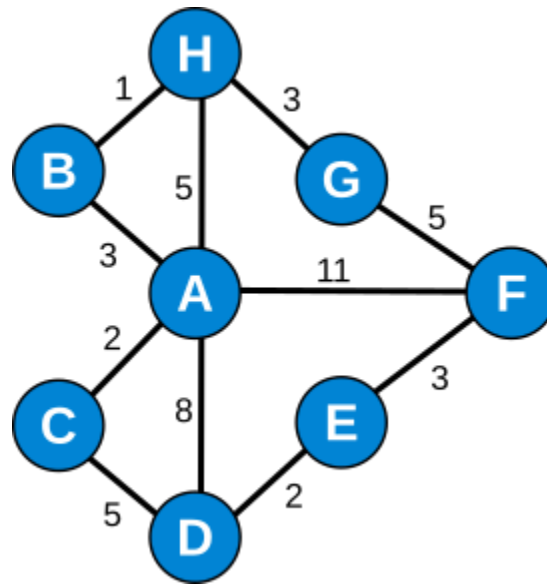
```
<8;"Mateo">
```

```
<6;"Ana">
```

```
<9;"Tomas">
```

```
<15;"Mariano">
```

- 4)
- A) Explique de qué se trata el algoritmo de **Floyd-Warshall**, para qué sirve y cómo funciona.
 - B) Aplique el algoritmo de **Dijkstra** al siguiente grafo, empezando por el vértice **A**. Justifique.



- 5) Implemente en python el **TDA Hash**. Debe ser de direccionamiento abierto (con o sin zona de desbordamiento). No hace falta implementar el TDA completo, pero se pide al menos las funciones **insertar** y **buscar**. Puede suponer que ya existe una función que rehashea y puede invocarla cuando crea necesario. Defina lo que haga falta y explique la solución propuesta.