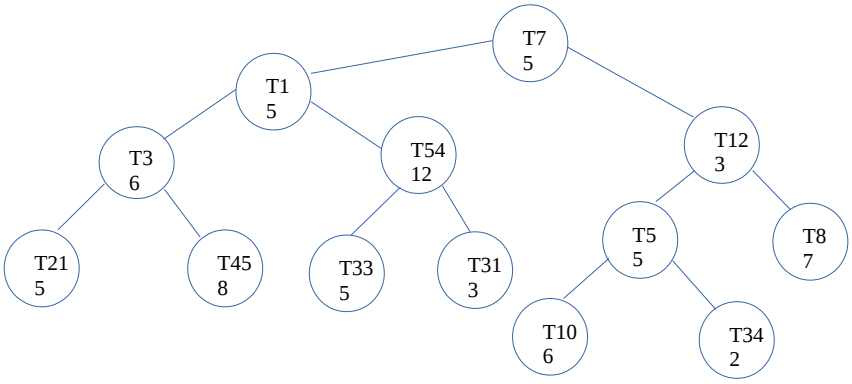
EJERCICIO 3.- Cálculo de coste (1,5 puntos)

Tenemos un árbol binario en el que cada nodo contiene 2 valores:

- Identificador de tarea
- Coste de esa tarea

Las tareas se organizan de manera jerárquica, de manera que cada nodo abarca todas las subtareas que forman parte de ella. Por ejemplo, la tarea T12 también incluye las subtareas T10, T5, T34 y T8.



Se pide **implementar** el subprograma coste*(idTarea)* que dirá el coste de una tarea (incluyendo también el coste de todas sus subtareas). **Calcular también el coste del subprograma**.

```
public class Tarea {
	String id;
	int coste;
}

public class BinaryTreeNode {
	Tarea element;
	BinaryTreeNode left, right;
}

public class ArbolTareas {

  BinaryTreeNode root;

  public int coste(String idTarea)

}
```

Por ejemplo:

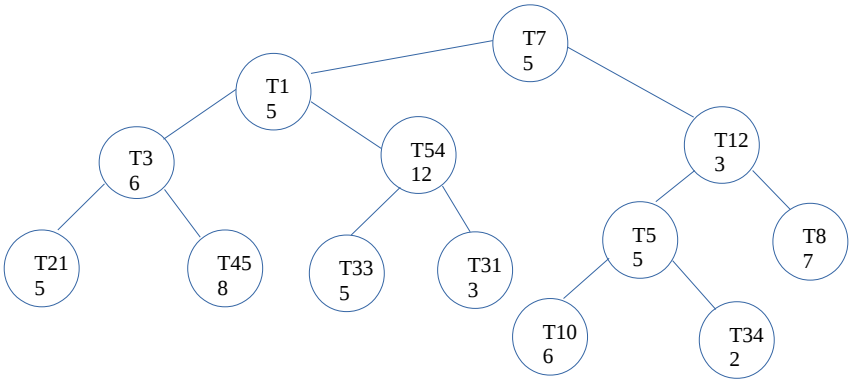
- coste(“T31”) → 3
- coste(“T12”) → 23
- coste(“T7”) → 72

ARIKETA 3.- Kostuaren kalkulua (1,5 puntu)

Zuhaitz bitar batean elementu bakoitzak bi balio ditu:

- Zereginaren identifikatzailea
- Zeregin horren kostua

Zereginak era hierarkikoan antolatzen dira, adabegi bakoitzak bere azpitik dauden azpizeregin guztiak hartzen dituelarik. Adibidez, T12 zereginak bere barne hartzen ditu T10, T5, T34 eta T8 zereginak ere.



kostua(zeregina) azpiprograma **inplementatu** behar da, zeregin baten kostua kalkulatzeko (zeregin horren barruan dauden azpizeregin guztiak barne). **Egindako azpiprogramaren kostua kalkulatu**.

```
public class Zeregin {
	String id;
	int kostua;
}

public class BinaryTreeNode {
	Zeregin element;
	BinaryTreeNode left, right;
}

public class ZereginenZuhaitza {

  BinaryTreeNode root;

  public int kostua(String idZeregin)

}
```

Adibidez:

- kostua(“T31”) → 3
- kostua(“T12”) → 23
- kostua(“T7”) → 72