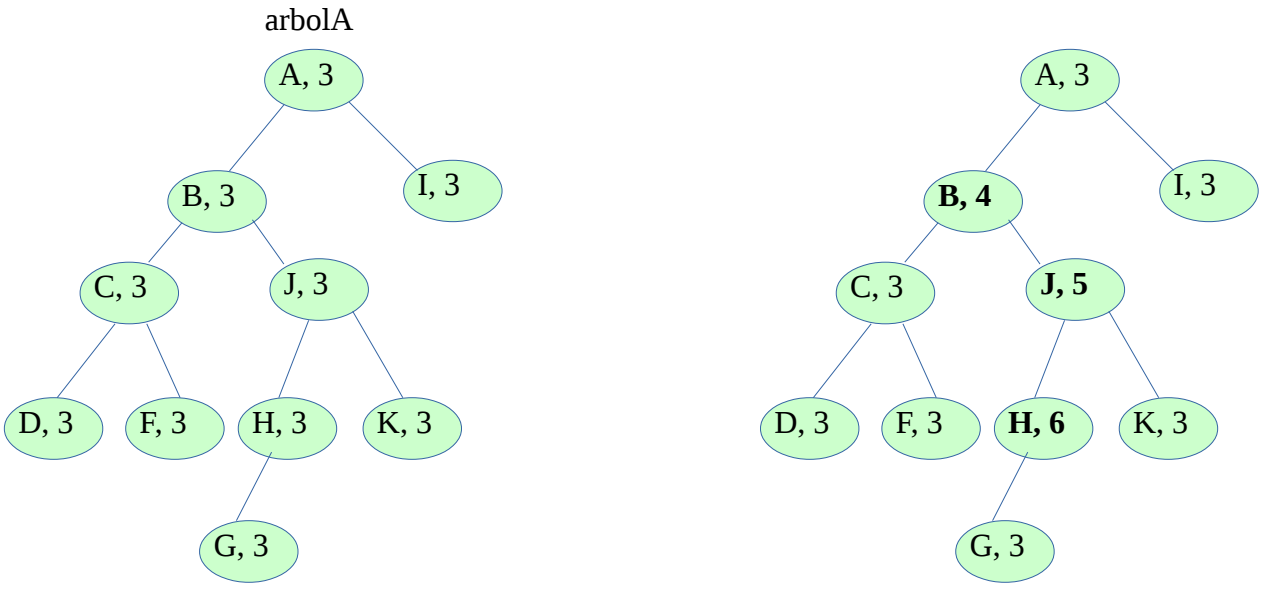
4. Árbol de juego (1,5 puntos)

Tenemos un árbol que representa un juego. Los jugadores se estructuran en forma de árbol binario, y cada jugador tiene una serie de puntos asignados.

```
public class Info {
    String   s;
    Integer  puntos;
}

public class Nodo {
    Info     content;
    Nodo     izq, der;
    Nodo     padre;
}

public class Arbol {
   private Nodo root;

   public void premiar(int puntos, String elem);
}
```

Cuando un jugador consigue puntuar, el juego establece que se sumarán n puntos a ese jugador, y además se irán repartiendo puntos entre los antecesores de ese jugador, de manera que a su antecesor directo se le asignarán (n-1) puntos, (n-2) a su antecesor, y así sucesivamente.

Por ejemplo, dado el árbol de la figura (a), la llamada a “arbolaA.premiar(3, “H”)” produciría el árbol de la figura (b), sumando 3 puntos al jugador “H”, 2 puntos a “J” y un punto a “B”.



(a) (b)

Se pide:

1. Implementar el método “premiar()”

2. Calcular, de manera razonada, el coste del algoritmo resultante.

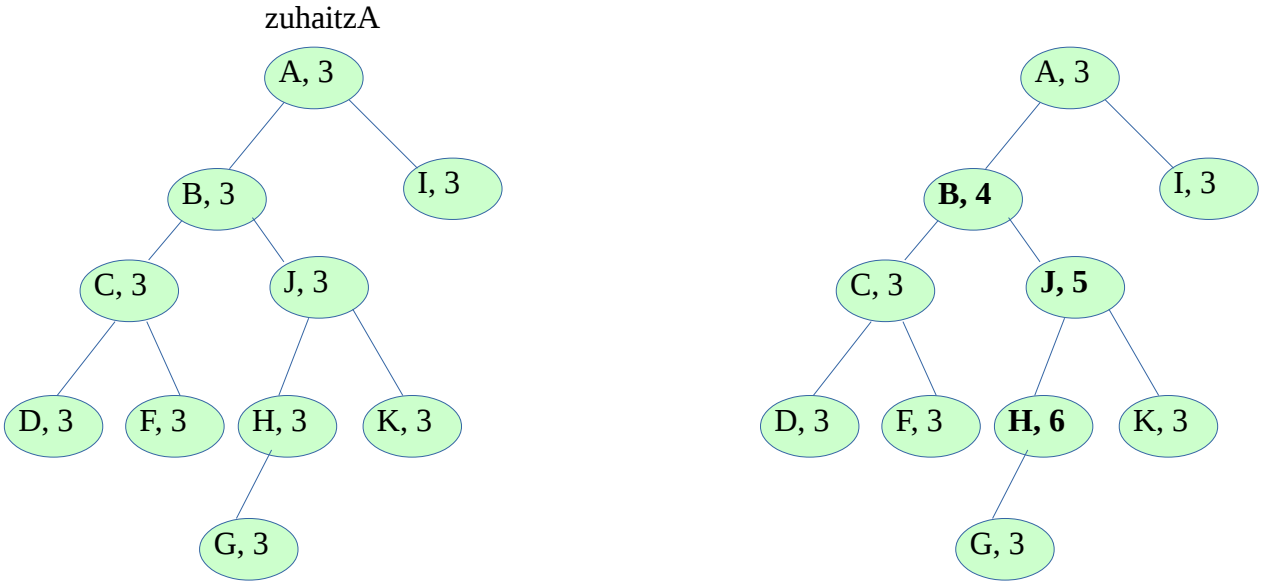
4. Jokoaren zuhaitza (1,5 puntu)

Joko bateko informazioa gordetzeko zuhaitz bitar bat dugu. Jokalari bakoitzak puntuazio bat dauka.

```
public class Info {
    String   s;
    Integer  puntuak;
}

public class Node {
    Info     content;
    Node     ezkerra, eskuina;
    Node     gurasoa;
}

public class Zuhaitza {
   private Node root;

   public void saritu(int puntuak, String elem);
}
```

Jokalari batek puntuak lortzen dituenean, jokoak esaten du jokalari horri n puntu emango zaizkiola, eta puntuak banatuko direla bere arbasoen artean: (n-1) puntu bere gurasoari, (n-2) hurrengoari eta abar.

Adibidez, (a) irudiko zuhaitzean “zuhaitzA.saritu(3, “H”)” egingo bagenu, (b) irudiko zuhaitza emango luke, H-ri 3 puntu, J-ri 2 puntu eta B-ri puntu bat emanez.



(a) (b)

Hau eskatzen da:

1. “**`saritu`**(...)” metodoaren inplementazioa

2. Kalkulatu, modu arrazoituan, algoritmoaren kostua.