```
import java.util.ArrayList;
import java.util.HashMap;

public class DrawTree {

//method to find the root of the index or -1
	public int findRoot(int[] parents) {

//initialization
		int rootIndex = 0;

//loop through parents
		for (int i = 0; i < parents.length; i++) {

//if the current index == -1
			if (parents[i] == -1) {

//update index of root
				rootIndex = i;
			}
		}

//return index of root
		return rootIndex;
	}

	//turns the hashmap into the drawing
	public ArrayList<String> hashmapToDraw(String preface, String root, int
depth, ArrayList<String> list,
			boolean sibling, HashMap<String, ArrayList<String>> map) {

//add a plus dash to every node
		String node = preface + "+-" + root;

//if there's a sibling add a connector
		if (sibling) {

			preface = preface + "| ";

//everything else
		} else {
```

```
//add the indentation
                preface = preface + "  ";
            }

//print the node
            System.out.println(node);

//add the node to the list
            list.add(node);

//if map contains the root
            if (map.containsKey(root)) {
                ArrayList<String> children = map.get(root);

//
                String lastChild = children.get(children.size() - 1);
                if (!children.isEmpty()) {

//loop through children
                    for (String child : children) {

//if the map contains the child and the child is not the last child
                        if (map.containsKey(child) & !child.equals(lastChild)) {

//it is a sibling
                            sibling = true;
//else its not
                        } else {
                            sibling = false;
                        }

                        list = hashmapToDraw(preface, child, depth + 1, list,
sibling, map);
                    }
                }
            }

//return the list
            return list;
        }
```

```
//
        public String[] illustrating(int[] parents, String[] names) {

//finds the index of the root (where the -1 is)
                int rootIndex = findRoot(parents);

//root is the index of -1
                String root = names[rootIndex];

//make a hash map with an string array list
                HashMap<String, ArrayList<String>> map = new HashMap<String,
ArrayList<String>>();

                map = recursiveDraw(root, parents, names, map);

// turn map into output
                ArrayList<String> output = new ArrayList<String>();
                output = hashmapToDraw("", root, 0, output, false, map);

//string array of the output
                String[] outputArray = output.toArray(new String[output.size()]);

                //return the output string array
                return outputArray;
        }

//draw the map recursively
        public HashMap<String, ArrayList<String>> recursiveDraw(String root, int[]
parents, String[] names,
                        HashMap<String, ArrayList<String>> map) {

//make a new array list for the children
                ArrayList<String> children = new ArrayList<String>();

// find all values whose parent == root
                for (int i = 0; i < names.length; i++) {

//if parents at i is not -1
                        if (parents[i] != -1) {

//
                                String parentName = names[parents[i]];
//if the name of the parent is the root
                        if (parentName.equals(root)) {

// add to list of children
                              children.add(names[i]);
                        }
                  }

            }

//if children is empty return the map
            if (children.isEmpty()) {
                  return map;

            } else {

//insert root and children into the map
                  map.put(root, children);

//loop through children
                  for (String child : children) {

//call the function and return map
                        map = recursiveDraw(child, parents, names, map);
                  }

//return the map
                  return map;
            }

      }



      /*
       * Citing
       * https://stackoverflow.com/questions/43346321/how-can-i-save-a-tree-
structure-in-a-hashmap
       * https://github.com/mcdickenson/data-algs-java/blob/master/APTs/Set6/
src/DrawTree.java
       *
       */
}
```