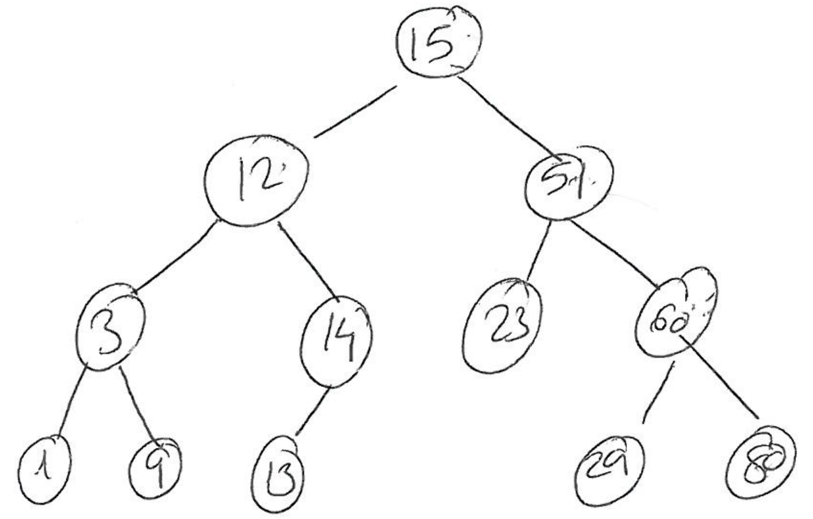
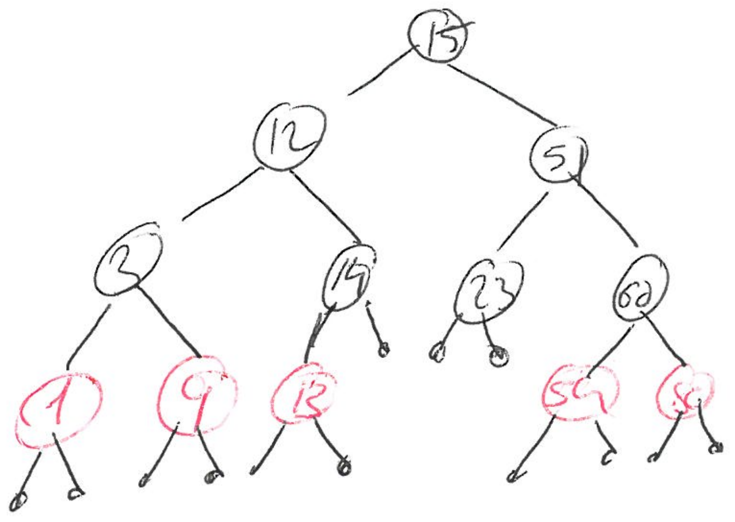
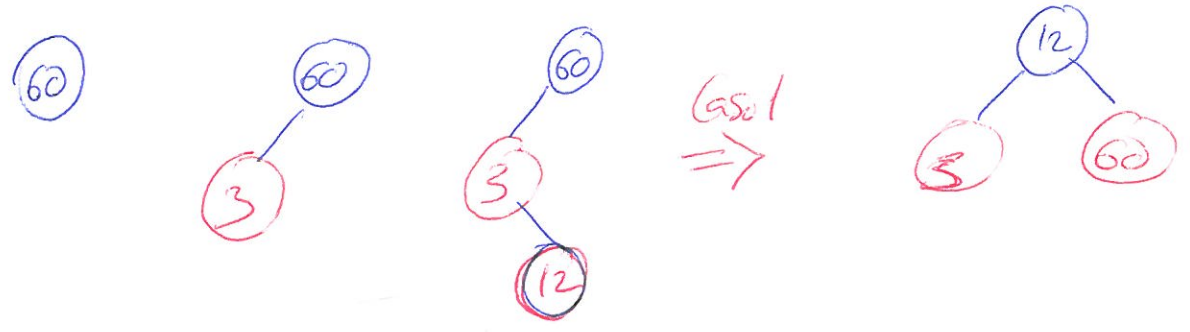
ercici 1a.

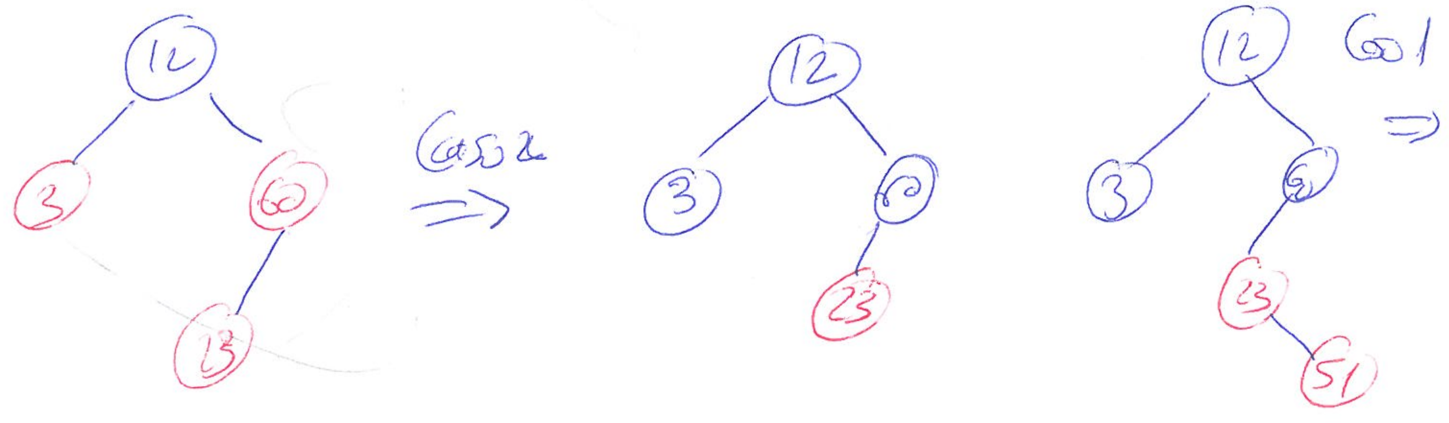
AVL

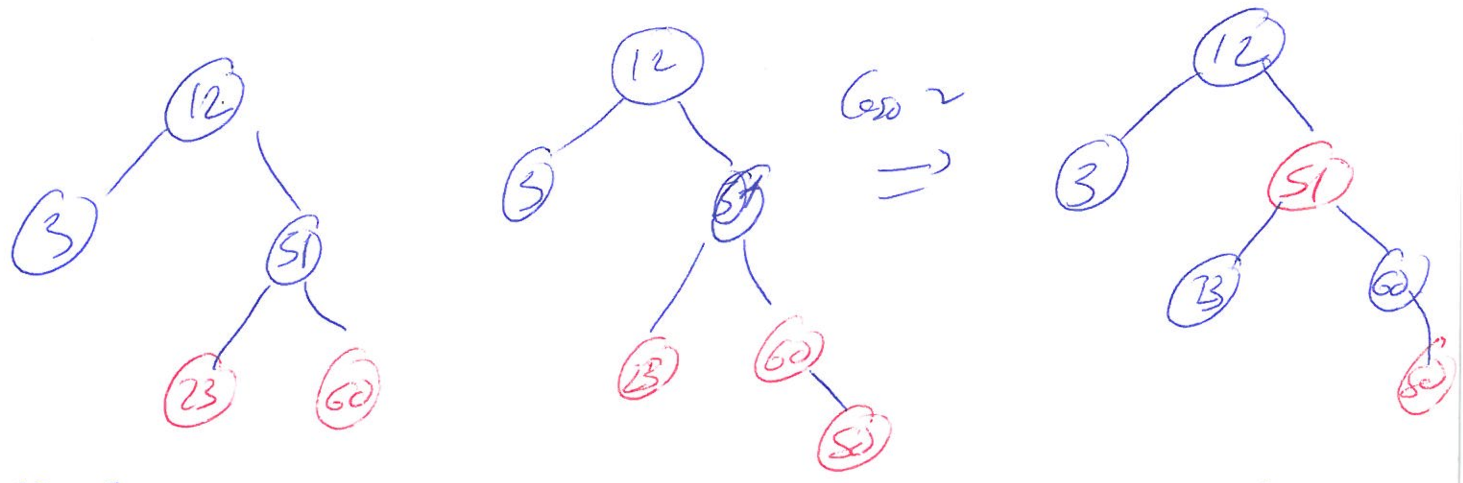


RN



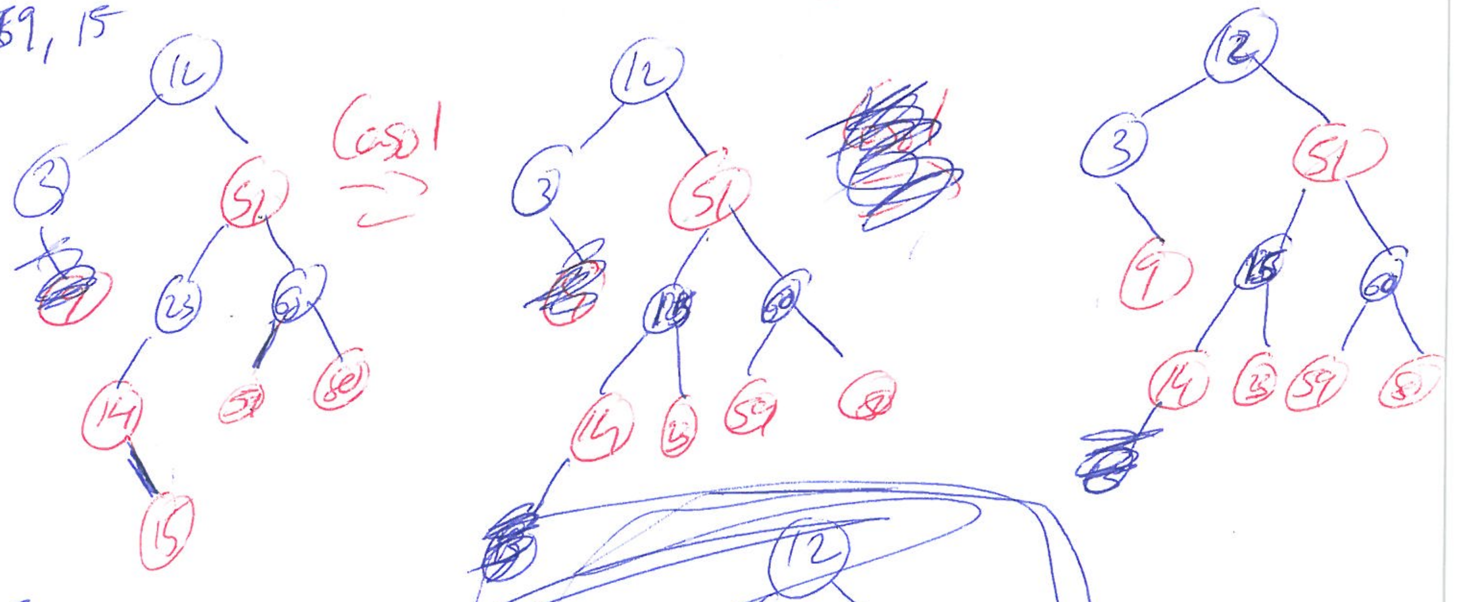






14, ~~59~~, 15

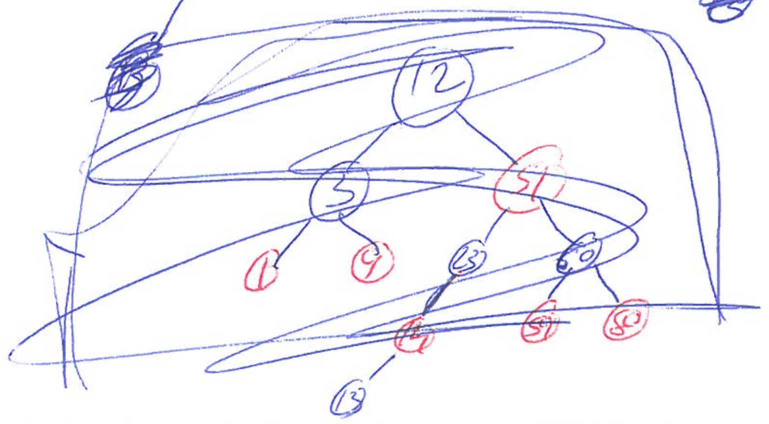
$\Rightarrow$

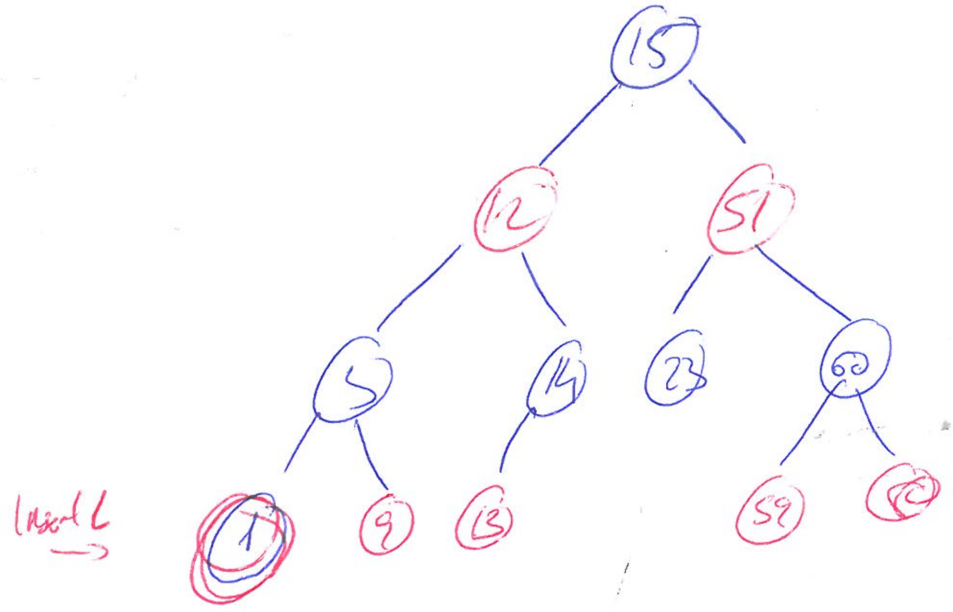


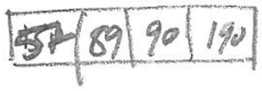
13

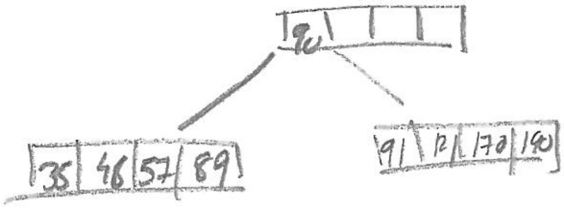
$\Rightarrow$

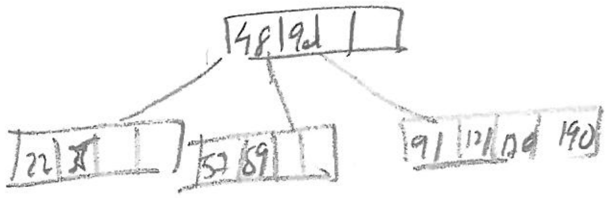
Caso 2

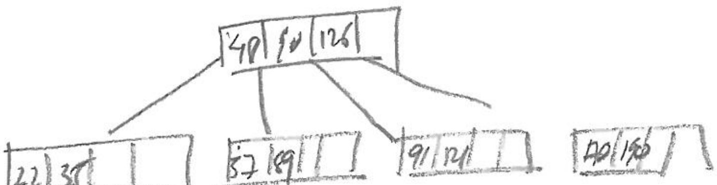


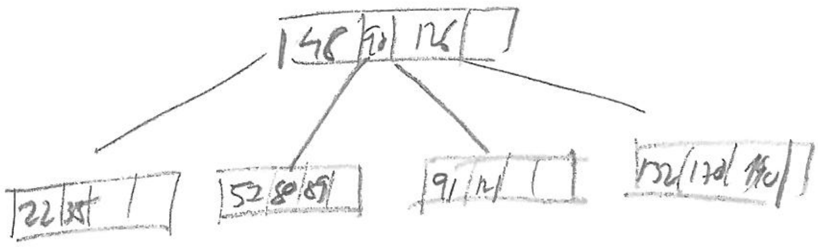

15
12
51
5
14
23
60
Insert L
1
9
13
59

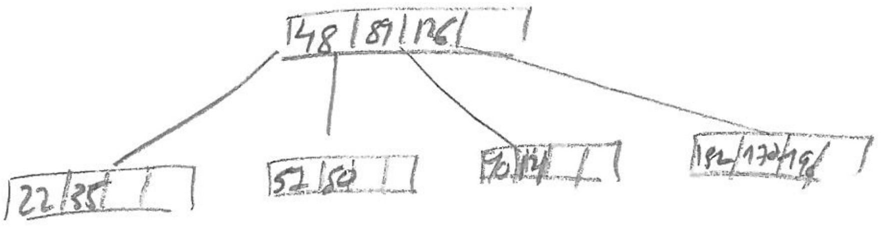

57 89 90 190

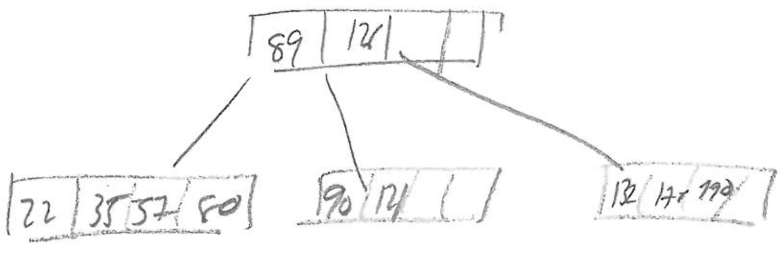

90
35 46 57 89
91 12 170 190


48 90
22 35
57 89
91 12 170 190

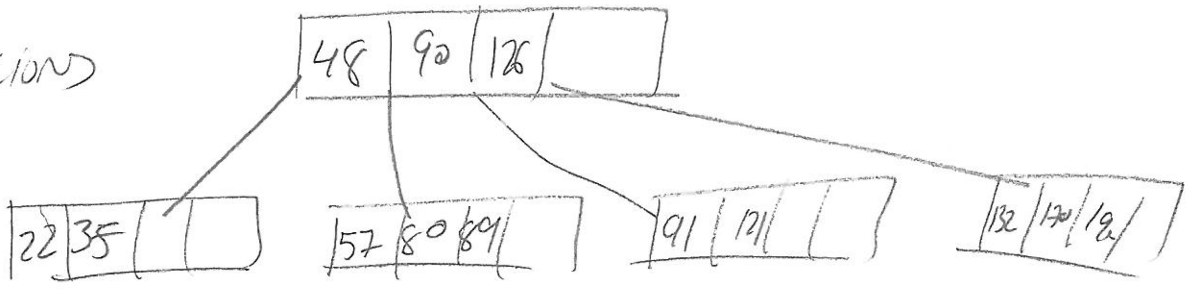

48 90 126
22 35
57 89
91 121
170 190


48 90 126
22 35
57 80 89
91 121
132 170 190

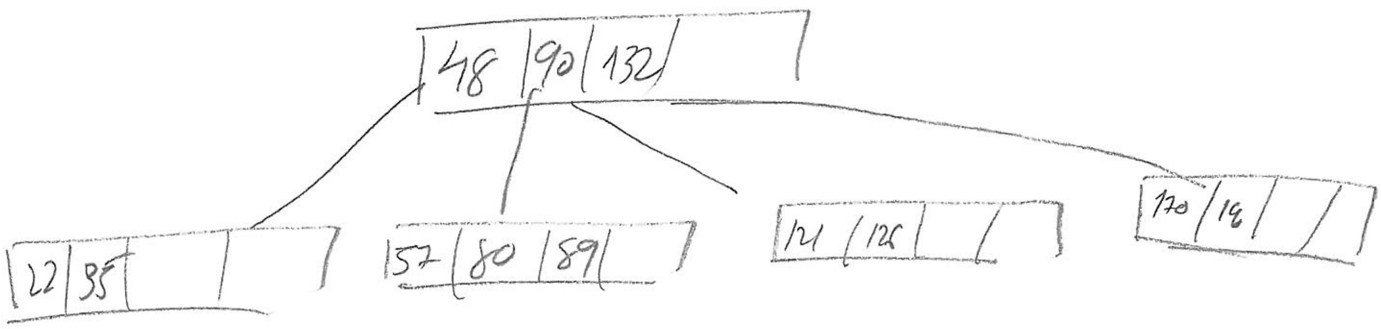

48 89 126
22 35
57 80
90 121
132 170 190


89 126
22 35 57 80
90 121
132 170 190

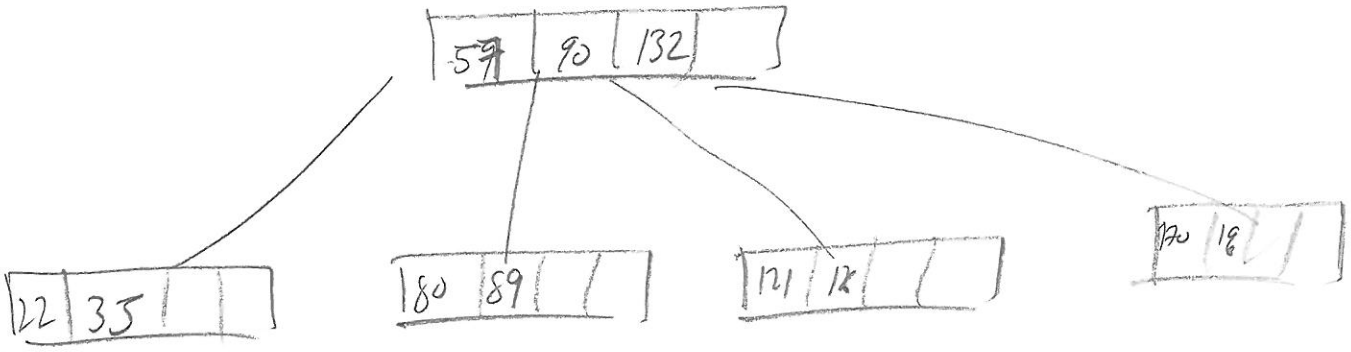
Insercions



Eliminar 91



Eliminar 48



```java
/*
 * To change this license header, choose License Headers in Project
     Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package examen;

import java.util.Iterator;
import material.tree.Position;
import material.tree.Tree;
import material.tree.iterator.TreeIteratorFactory;

/**
 *
 * @author jesussanchezoro
 */
public class InternalNodeIteratorFactory<E> implements TreeIteratorFactory<E>
    {

    @Override
    public Iterator<Position<E>> createIterator(Tree<E> tree) {
        return new InternalNodeIterator<>(tree);
    }

    @Override
    public Iterator<Position<E>> createIterator(Tree<E> tree, Position<E> pos)
        {
        return new InternalNodeIterator<>(tree, pos);
    }

}
```

```java
/*
 * To change this license header, choose License Headers in Project
     Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package examen;

import java.util.ArrayDeque;
import java.util.Iterator;
import java.util.Queue;
import material.tree.Position;
import material.tree.Tree;

/**
 *
 * @author jesussanchezoro
 */
public class InternalNodeIterator<E> implements Iterator<Position<E>> {

    private final Queue<Position<E>> nodeQueue;
    private final Tree<E> tree;

    public InternalNodeIterator(Tree<E> tree) {
        nodeQueue = new ArrayDeque<>();
        this.tree = tree;
        Position<E> root = tree.root();
        if (tree.isInternal(root)) {
            nodeQueue.add(root);
        }
    }

    public InternalNodeIterator(Tree<E> tree, Position<E> pos) {
        nodeQueue = new ArrayDeque<>();
        this.tree = tree;
        if (tree.isInternal(pos)) {
            nodeQueue.add(pos);
        }
    }

    @Override
    public boolean hasNext() {
        return !nodeQueue.isEmpty();
    }

    @Override
    public Position<E> next() {
        Position<E> next = nodeQueue.poll();
        for (Position<E> child : tree.children(next)) {
            if (tree.isInternal(child)) {
                nodeQueue.add(child);
            }
        }
        return next;
    }



}
```

```java
/*
 * To change this license header, choose License Headers in Project
     Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package examen;

import java.util.Iterator;
import material.tree.Position;
import material.tree.binarytree.BinaryTree;
import material.tree.binarytree.LinkedBinaryTree;

/**
 *
 * @author jesussanchezoro
 */
public class Ejercicio2<E> {

    public boolean isPerfect(BinaryTree<E> t) {
        Iterator<Position<E>> it = t.iterator();
        while (it.hasNext()) {
            Position<E> p = it.next();
            if (t.isInternal(p) && (!t.hasLeft(p) || !t.hasRight(p))) {
                return false;
            }
        }
        return true;
    }

    public boolean isOdd(BinaryTree<E> t) {
        if (t.isEmpty() || (t.size() == 1)) {
            return true;
        }
        int descendants = isOddRec(t, t.root());
        return descendants >= 0;
    }

    private int isOddRec(BinaryTree<E> t, Position<E> pos) {
        int left = t.hasLeft(pos)?isOddRec(t, t.left(pos)):0;
        if (left < 0) {
            return -1;
        }
        int right = t.hasRight(pos)?isOddRec(t, t.right(pos)):0;
        if (right < 0) {
            return -1;
        }
        if ((left < 0) || (right < 0) || (left < ((left+right)/2)) ) {
            return -1;
        } else {
            return left+right+1;
        }
    }

    public static void main(String[] args) {
        Ejercicio2<Integer> ej2 = new Ejercicio2<>();
        BinaryTree<Integer> t = new LinkedBinaryTree<>();
        Position<Integer> pos1 = t.addRoot(1);
        Position<Integer> pos2 = t.insertLeft(pos1, 2);
        Position<Integer> pos3 = t.insertRight(pos1, 3);
        Position<Integer> pos4 = t.insertLeft(pos2, 4);
        Position<Integer> pos5 = t.insertRight(pos2, 5);

        System.out.println(ej2.isPerfect(t));

        BinaryTree<Integer> t2 = new LinkedBinaryTree<>();
        Position<Integer> pos6 = t2.addRoot(6);
        Position<Integer> pos7 = t2.insertLeft(pos6, 7);
        Position<Integer> pos8 = t2.insertRight(pos6, 8);
        Position<Integer> pos9 = t2.insertLeft(pos7, 9);
        Position<Integer> pos10 = t2.insertLeft(pos9, 10);

        System.out.println(ej2.isOdd(t2));
    }
}
```

```java
/*
 * To change this license header, choose License Headers in Project
     Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package examen.ejercicio3;

/**
 *
 * @author jesussanchezoro
 */
public class Main {

    public static void main(String[] args) {
        URJCInvest invest = new URJCInvest();
        OrganizationChart urjc = new OrganizationChart("URJC");
        Employee fs = new Employee("URJC", "Fernando Su·rez", "Dean", "");
        Employee ad = new Employee("URJC", "Abraham Duarte", "Full professor",
            "");
        Employee as = new Employee("URJC", "Antonio Sanz", "Full professor",
            "");
        Employee js = new Employee("URJC", "Jes˙s S·nchez-Oro", "Associate
            professor", "");
        Employee dc = new Employee("URJC", "David Concha", "Associate
            professor", "");
        urjc.addEmployee(null, fs);
        urjc.addEmployee("Fernando Su·rez", ad);
        urjc.addEmployee("Fernando Su·rez", as);
        urjc.addEmployee("Abraham Duarte", js);
        urjc.addEmployee("Antonio Sanz", dc);
        invest.addCompany(urjc);

        System.out.println("JEFES DE JESUS");
        for (Employee emp : invest.getChiefs("URJC", "Jes˙s S·nchez-Oro")) {
            System.out.println(emp);
        }
        System.out.println("JEFES DE ANTONIO");
        for (Employee emp : invest.getChiefs("URJC", "Antonio Sanz")) {
            System.out.println(emp);
        }
        System.out.println("JEFES DE FERNANDO");
        for (Employee emp : invest.getChiefs("URJC", "Fernando Su·rez")) {
            System.out.println(emp);
        }

        System.out.println("BECARIOS URJC");
        for (Employee emp : invest.getGrantHolders("URJC")) {
            System.out.println(emp);
        }

        System.out.println("FULL PROFESSORS");
        for (Employee emp : invest.getEmployees("Full professor")) {
            System.out.println(emp);
        }
        System.out.println("ASSOCIATE PROFESSORS");
        for (Employee emp : invest.getEmployees("Associate professor")) {
            System.out.println(emp);
        }
        System.out.println("DEANS");
        for (Employee emp : invest.getEmployees("Dean")) {
            System.out.println(emp);
        }
    }
}
```

```java
/*
 * To change this license header, choose License Headers in Project
     Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package examen.ejercicio3;

import java.util.ArrayList;
import java.util.List;
import material.maps.HashTableMapLP;
import material.maps.Map;

/**
 *
 * @author jesussanchezoro
 */
public class URJCInvest {

    private Map<String, OrganizationChart> companies;

    public URJCInvest() {
        companies = new HashTableMapLP<>();
    }

    public void addCompany(OrganizationChart company) {
        companies.put(company.getName(), company);
    }

    public OrganizationChart searchCompany(String name) throws
        IllegalStateException {
        OrganizationChart company = companies.get(name);
        if (company == null) {
            throw new IllegalStateException("The company does not belong to
                URJCInvest group");
        }
        return company;
    }

    public Iterable<Employee> getGrantHolders(String companyName) throws
        IllegalStateException {
        OrganizationChart company = searchCompany(companyName);
        return company.getGrantHolders();
    }

    public Iterable<Employee> getChiefs(String companyName, String employee)
        throws IllegalStateException {
        OrganizationChart company = searchCompany(companyName);
        return company.getChiefs(employee);
    }

    public Iterable<Employee> getEmployees(String position) {
        List<Employee> empWithPos = new ArrayList<>();
        for (OrganizationChart chart : companies.values()) {
            empWithPos.addAll(chart.getEmployees(position));
        }
        return empWithPos;
    }
}
```

```java
/*
 * To change this license header, choose License Headers in Project
     Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package examen.ejercicio3;

/**
 *
 * @author jesussanchezoro
 */
public class Employee {
    private String company;
    private String name;
    private String position;
    private String description;

    public Employee(String company, String name, String position, String
        description) {
        this.company = company;
        this.name = name;
        this.position = position;
        this.description = description;
    }

    public String getCompany() {
        return company;
    }

    public String getDescription() {
        return description;
    }

    public String getName() {
        return name;
    }

    public String getPosition() {
        return position;
    }

    @Override
    public String toString() {
        return name+" ("+position+" at "+company+")";
    }


}
```

```java
/*
 * To change this license header, choose License Headers in Project
     Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package examen.ejercicio3;

import java.util.ArrayList;
import java.util.List;
import material.tree.LinkedTree;
import material.tree.Position;

/**
 *
 * @author jesussanchezoro
 */
public class OrganizationChart {

    private LinkedTree<Employee> chart;
    private String name;

    public OrganizationChart(String name) {
        this.name = name;
        chart = new LinkedTree<>();
    }

    public String getName() {
        return name;
    }

    public void addEmployee(String parent, Employee e) {
        if (parent == null) {
            chart.addRoot(e);
        } else {
            chart.add(e, findEmployee(parent));
        }
    }

    public Iterable<Employee> getGrantHolders() {
        List<Employee> grantHolders = new ArrayList<>();
        for (Position<Employee> emp : chart) {
            if (chart.isLeaf(emp)) {
                grantHolders.add(emp.getElement());
            }
        }
        return grantHolders;
    }

    private Position<Employee> findEmployee(String name) throws
        IllegalStateException {
        for (Position<Employee> emp : chart) {
            if (emp.getElement().getName().equalsIgnoreCase(name)) {
                return emp;
            }
        }
        throw new IllegalStateException("The employee "+name+" does not belong
            to the company");
    }
    public Iterable<Employee> getChiefs(String employee) throws
        IllegalStateException {
        List<Employee> chiefs = new ArrayList<>();
        Position<Employee> emp = findEmployee(employee);
        if (!chart.isRoot(emp)) {
            while (!chart.isRoot(emp)) {
                chiefs.add(chart.parent(emp).getElement());
                emp = chart.parent(emp);
            }
        }
        return chiefs;
    }

    public List<Employee> getEmployees(String position) {
        List<Employee> empWithPos = new ArrayList<>();
        getEmployeesRec(chart.root(), position, empWithPos);
        return empWithPos;
    }

    private void getEmployeesRec(Position<Employee> node, String position,
        List<Employee> empWithPos) {
        if (node.getElement().getPosition().equalsIgnoreCase(position)) {
            empWithPos.add(node.getElement());
        } else {
            for (Position<Employee> child : chart.children(node)) {
                getEmployeesRec(child, position, empWithPos);
            }
        }
    }
}
```