A white paper with black text

Description automatically generated

A paper with text and a diagram

Description automatically generated

A screenshot of a math problem

Description automatically generated

A screenshot of a computer

Description automatically generated

import java.util.Vector;

public class OptimalBST {

    static Vector<Vector<Double>> e;

    static Vector<Vector<Double>> root;

    public static void optimal\_bst(Vector<Double> p, Vector<Double> q, Integer n) {

        e = new Vector<Vector<Double>>();

        root = new Vector<Vector<Double>>();

        for (int i = 0; i < n + 1; i++) {

            Vector<Double> tempE = new Vector<Double>();

            Vector<Double> tempRoot = new Vector<Double>();

            for (int j = 0; j < n + 1; j++) {

                tempE.add(0.0);

                tempRoot.add(0.0);

            }

            e.add(tempE);

            root.add(tempRoot);

        }

        for (int i = 1; i < n + 2; i++) {

            e.elementAt(i - 1).set(i - 1, q.elementAt(i - 1));

            root.elementAt(i - 1).set(i - 1, (double)(i - 1));

        }

        for (int l = 1; l < n + 1; l++) {

            for (int i = 1; i < n - l + 2; i++) {

                int j = i + l - 1;

                e.elementAt(i - 1).set(j, Double.MAX\_VALUE);

                double w = 0.0;

                for (int r = i; r < j + 1; r++) {

                    w += p.elementAt(r) + q.elementAt(r);

                }

                for (int r = i; r < j + 1; r++) {

                    double t = e.elementAt(i - 1).elementAt(r - 1) + e.elementAt(r).elementAt(j) + w;

                    // Allow a small tolerance for floating-point comparisons

                    if (t - e.elementAt(i - 1).elementAt(j) < 0.000001) {

                        e.elementAt(i - 1).set(j, t);

                        root.elementAt(i - 1).set(j, (double)r);

                    }

                }

            }

        }

    }

    public static void main(String[] args) {

        Vector<Double> p = new Vector<Double>();

        Vector<Double> q = new Vector<Double>();

        Integer n = 5;

        p.add(0.0);

        p.add(0.15);

        p.add(0.10);

        p.add(0.05);

        p.add(0.10);

        p.add(0.20);

        q.add(0.05);

        q.add(0.10);

        q.add(0.05);

        q.add(0.05);

        q.add(0.05);

        q.add(0.10);

        optimal\_bst(p, q, n);

        for (int i = 0; i < e.size(); i++) {

            System.out.println(e.elementAt(i));

        }

        for (int i = 0; i < root.size(); i++) {

            System.out.println(root.elementAt(i));

        }

    }

}

A computer screen with white text and numbers

Description automatically generated