

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА № 52

ОТЧЕТ ЗАЩИЩЕН С ОЦЕНКОЙ _____

ПРЕПОДАВАТЕЛЬ

Доцент, канд. тех. наук
должность, уч. степень, звание

подпись, дата

Марковская Н.В.
инициалы, фамилия

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

Вероятностный расчёт на основе случайного графа

по курсу: Надежность инфокоммуникационных систем

СТУДЕНТ ГР. № 5912
номер группы

25.02.2022
подпись, дата

Нам Д. О.
инициалы, фамилия

Санкт-Петербург
2022

1. Цель работы

В случайном графе вычислить вероятность существования пути между заданной парой вершин. Построить зависимость вероятности существования пути в случайном графе от вероятности существования ребра.

2. Задание

На рисунке 1 изображён случайный граф. Ищем вероятность существования пути из вершины 1 в вершину 6. $P_1 = P_2 = \dots = P_9$

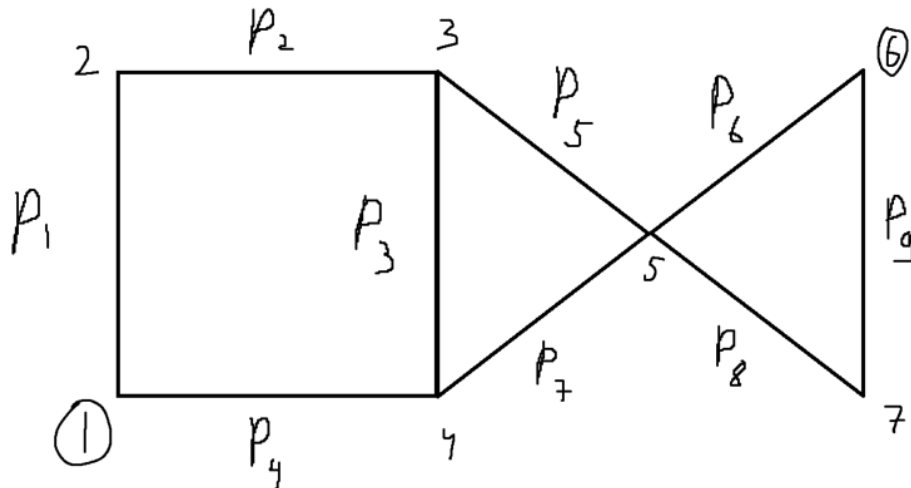
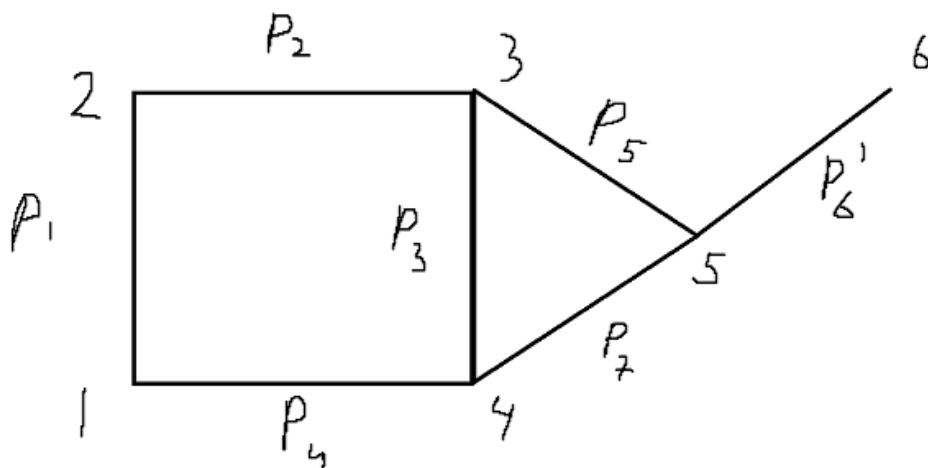


Рисунок 1. Случайный граф

3. Выполнение задания

Для нахождения вероятности существования пути 1-6 нужно провести упрощение структуры и декомпозицию случайного графа.

$$P'_6 = P_6 + P_8 \cdot P_9 - P_6 \cdot P_8 \cdot P_9$$



$$P_1' = P_1 \cdot P_2$$

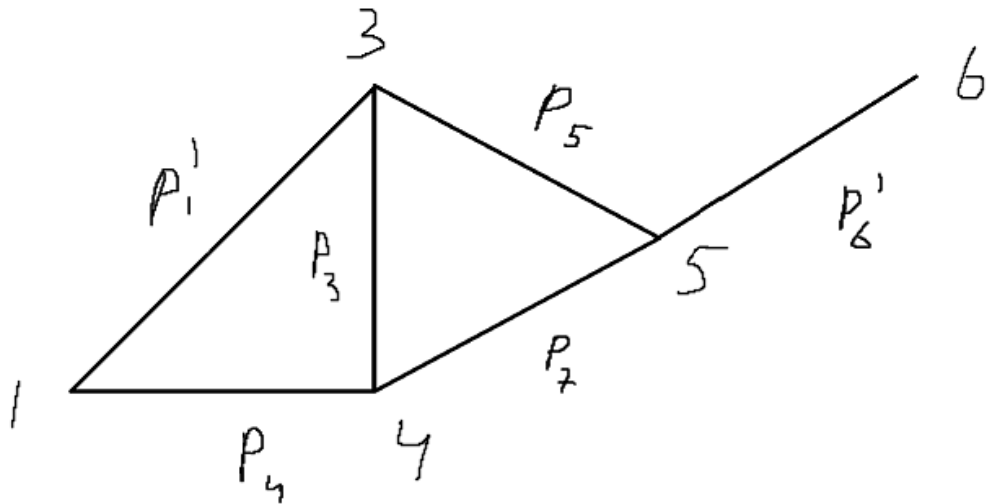
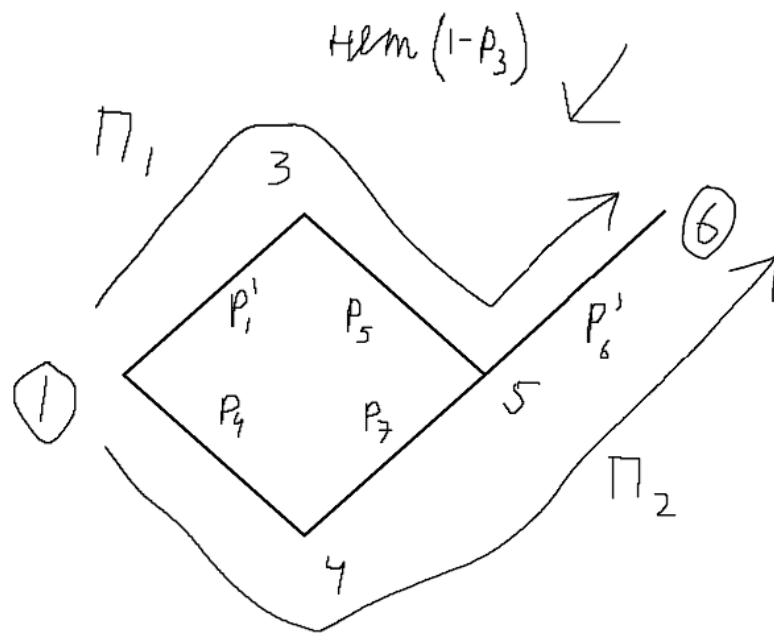
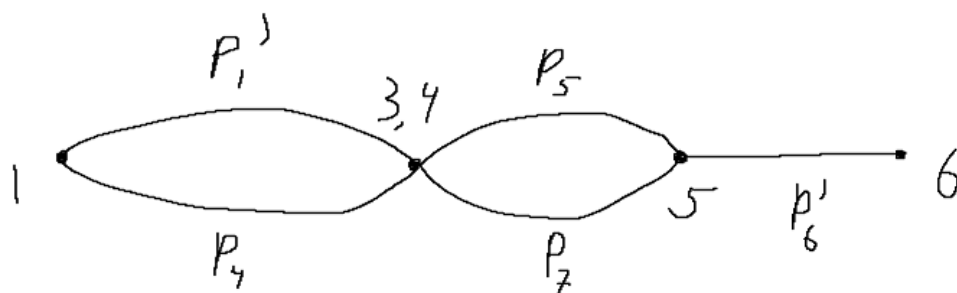


Рисунок 2. Упрощение структуры графа



$$\Pr \{n \text{ 1,6} \mid \text{лем 3,4}\} = P_1' P_5 P_6' + P_4 P_7 P_6' - \\ - P_1' P_5 P_4 P_7 P_6'$$



$$P_r \{ n_{1,6} \mid \text{есть } 3,4 \} = (P_1' + P_4 - P_1' P_4) \cdot (P_5 + P_7 - P_5 P_7) \cdot P_6'$$

Рисунок 3. Декомпозиция графа

Итоговая формула вероятности пути из вершины 1 в вершину 6:

$$P_r \{ n_{1,6} \} = P_r \{ n_{1,6} \mid \text{нет } 3,4 \} \cdot (1 - P_3) + P_r \{ n_{1,6} \mid \text{есть } 3,4 \} \cdot P_3$$

Используя приведенную выше формулу вычисляем зависимость вероятности существования пути от вероятности существования ребра.

Р ребра	Р пути по формуле	Р пути через полный перебор
0	0	0
0.1	0.001303858	0.001303858
0.2	0.012724736	0.012724736
0.3	0.049272894	0.049272894
0.4	0.1265950719	0.1265950719
0.5	0.25390625	0.25390625
0.6	0.4272583679	0.4272583679
0.7	0.625948246	0.625948246
0.8	0.814383104	0.814383104
0.9	0.9505124819	0.9505124819
1	1	1

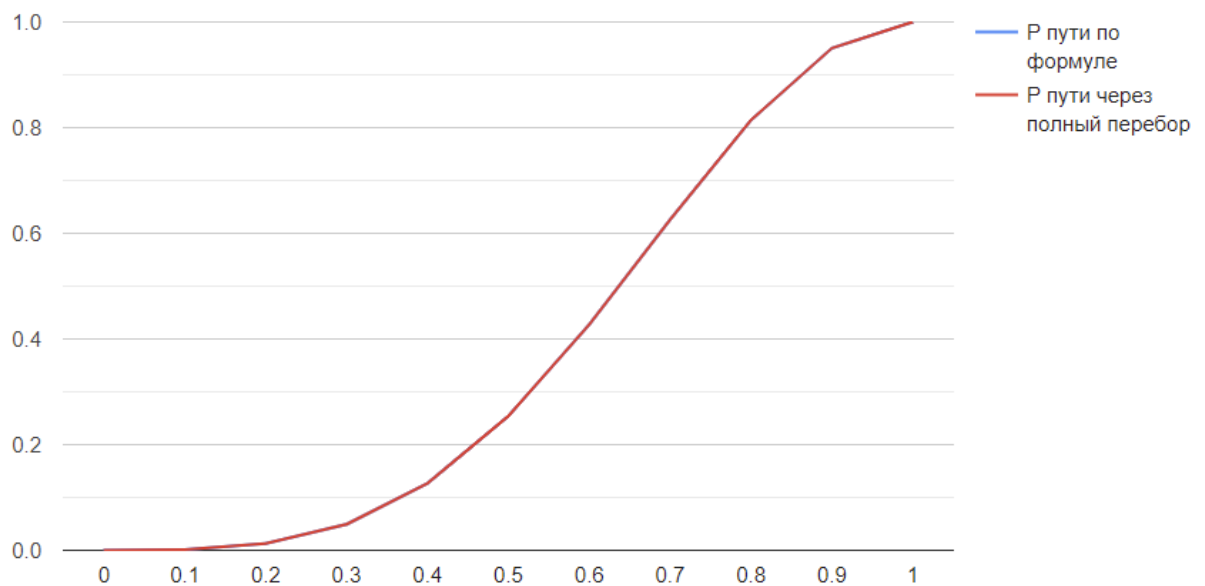


График 1. Зависимость P пути от P ребра
в заданном случайном графе

Как видно из графика 1 итоговая формула оказалась верной, т.к. результат вычисления по ней полностью совпал с результатом полного перебора.

Полный перебор был реализован на языке Java с помощью кода Грея. Поскольку в подграфе у каждого ребра есть всего два состояния (1 – есть в подграфе, 0 – нет в подграфе), то код Грея идеально подходит для полного перебора.

Например, если код Грея равен «000101100», то формула вероятности пути из вершины 1 в вершину 6 будет выглядеть как $P = p^3 * (1 - p)^6$. Перед тем как считать эту вероятность для данного подграфа нужно проверить собственно существование пути из 1 в 6. Для этого используется обычный алгоритм DFS. В приложении А можно ознакомиться с main-ом программы.

4. Вывод

В ходе работы был упрощён случайный граф, выведена формула вероятности существования пути между заданной парой вершин, правильность которой была подтверждена результатами программного полного перебора всех возможных подграфов случайного графа.

```

public static void main(String args[]) {
    int n = 7; //кол-во вершин
    int l = 9; //кол-во рёбер
    int from = 0; //начальная вершина
    int find = 5; //конечная вершина
    double p = 0.7; //вероятность сущ ребра
    LinkedList<Pair> lp = new LinkedList<>();
    lp.add(new Pair(0, 1)); //p1
    lp.add(new Pair(0, 3)); //p2
    lp.add(new Pair(1, 2)); //p3
    lp.add(new Pair(2, 3)); //p4
    lp.add(new Pair(2, 4)); //p5
    lp.add(new Pair(3, 4)); //p6
    lp.add(new Pair(4, 5)); //p7
    lp.add(new Pair(4, 6)); //p8
    lp.add(new Pair(5, 6)); //p9

    Graph subGraph = new Graph(n);

    subGraph.addVertex('A'); //1
    subGraph.addVertex('B'); //2
    subGraph.addVertex('C'); //3
    subGraph.addVertex('D'); //4
    subGraph.addVertex('E'); //5
    subGraph.addVertex('F'); //6
    subGraph.addVertex('G'); //7

    Gray grayCode = new Gray(l); //от кол-ва рёбер зависит длина кода Грея

    int count = 0;
    int trueCount = 0;

    double sum = 0;
    for (int i = 0; i < grayCode.code.size(); ++i) {
        LinkedList<String> tmp = grayCode.code.get(i); //0 0 0 0 0 0 0 0 0

        for (int j = 0; j < l; ++j) {
            if (tmp.get(j).equals("1")) {
                subGraph.addEdge(lp.get(j).f, lp.get(j).s);
                trueCount++;
            }
        }

        if (subGraph.dfs(from, find)) {
            count++;
            sum += Math.pow(p, trueCount) * Math.pow((1-p), (l-trueCount));
        }

        //del
        subGraph.clear();
        trueCount = 0;
    }

    System.out.println();
    System.out.println("Всего " + grayCode.code.size() + " подграфов");
    System.out.println("Из них в " + count + " есть путь от " + ++from + " до " +
++find);
    System.out.println("Вероятность равна: " + sum + " при p равном " + p);
}

```