

Министерство образования и науки Российской Федерации  
Санкт-Петербургский политехнический университет Петра Великого  
Кафедра прикладной математики и информатики

Отчет по лабораторной работе №2 на тему  
"Алгоритм поиска наибольшего паросочетания"  
по дисциплине  
**"Дискретная математика"**

Выполнил студент гр. 5030102/20201 Фатахов Т.М.

Санкт-Петербург  
2024

# 1 Задание

Требовалось реализовать алгоритм поиска наибольшего паросочетания в двудольном графе. Алгоритм основан на поиске в глубину и нахождении увеличивающих цепей. Паросочетание в графе должно быть представлено массивом, где для каждой вершины второй доли хранится связанная с ней вершина из первой доли. Алгоритм должен работать с невзвешенными двудольными графами.

## 2 Язык программирования

Реализация выполнена на языке Python версии 3.9. Python был выбран за счёт своей простоты, а также возможности быстрого тестирования и реализации графовых алгоритмов. Использование объектно-ориентированного подхода позволило структурировать программу в виде классов, что делает её легко расширяемой.

## 3 Описание алгоритма

Алгоритм основан на поиске увеличивающей цепи в двудольном графе с помощью поиска в глубину. Для каждой вершины первой доли графа пытаемся найти путь, который увеличивает текущее паросочетание, и перестроить его.

Основные шаги алгоритма:

1. Инициализация массива паросочетания  $M$ , где  $M[u] = 0$  для всех вершин  $u$  второй доли  $V_2$ . Это означает, что в начале работы паросочетание пусто.
2. Для каждой вершины  $v$  из первой доли  $V_1$  запускаем поиск в глубину с целью нахождения увеличивающей цепи.
3. Если для вершины  $u$  из второй доли либо нет паросочетания, либо можно перестроить текущее паросочетание, обновляем массив  $M$ , добавляя в него новое ребро.
4. Процесс повторяется для всех вершин первой доли, пока не будут обработаны все вершины и не удастся найти новые увеличивающие цепи.

Функция поиска увеличивающей цепи ('Aug') работает следующим образом:

1. Если вершина  $v$  уже была посещена в текущем вызове поиска, то возвращаем **False** — увеличивающая цепь не найдена.
2. Иначе помечаем вершину  $v$  как посещённую.
3. Для каждой смежной вершины  $u$  второй доли пытаемся либо присоединить её к текущему паросочетанию, либо найти увеличивающую цепь, начиная с её текущего партнёра в паросочетании  $M[u]$ .
4. Если удалось найти увеличивающую цепь, обновляем паросочетание и возвращаем **True**, иначе — **False**.

## 4 Демонстрация работы алгоритма

Рассмотрим работу алгоритма на простом примере.

Дан двудольный граф  $G = (V_1, V_2, E)$ , где:

$$V_1 = \{1, 2, 3\}, \quad V_2 = \{1, 2, 3\}$$

$$E = \{(1, 1), (1, 2), (1, 3), (2, 1), (2, 2), (3, 1)\}$$

Граф можно представить в виде таблицы смежности:

	1	2	3
1	1	1	1
2	1	1	0
3	1	0	0

Table 1: Таблица смежности графа

Алгоритм поиска наибольшего паросочетания выполняется следующим образом:

- На первой итерации вершина 1 первой доли  $V_1$  соединяется с вершиной 1 второй доли  $V_2$ , образуя паросочетание  $\{(1, 1)\}$ .
- На второй итерации вершина 2 первой доли соединяется с вершиной 2 второй доли, образуя паросочетание  $\{(1, 1), (2, 2)\}$ .
- На третьей итерации вершина 3 первой доли соединяется с вершиной 1 второй доли, после чего алгоритм перестраивает паросочетание, и финальный результат будет  $\{(2, 1), (1, 2), (1, 3)\}$ .

Таким образом, максимальное паросочетание:  $\{(2, 1), (1, 2), (1, 3)\}$ .

## 5 Область применения и возможные ошибки

Данный алгоритм применяется в задачах нахождения наибольшего паросочетания в двудольных графах. Это может использоваться в задачах распределения ресурсов, задачах оптимального назначения и других.

Программа может дать сбой в следующих случаях:

- Граф содержит цикл или не является двудольным — в таких случаях программа может не дать корректный результат, так как алгоритм предназначен только для двудольных графов.
- Если входные данные не соответствуют ожидаемому формату, например, при отсутствии рёбер или неправильном формате списка смежности.

Программа не даст сбой при работе с корректными двудольными графами, поскольку для них алгоритм корректно находит максимальное паросочетание.

## 6 Формат входных и выходных данных

- Входные данные: граф передаётся в виде списка смежности, где ключ — вершина первой доли, а значение — список вершин второй доли, с которыми она соединена.
- Выходные данные: программа выводит паросочетание в виде списка, где для каждой вершины второй доли указана соответствующая вершина первой доли.

Пример входных данных для графа:

$$\{1 : [1, 2, 3], 2 : [1, 2], 3 : [1]\}$$

Пример выходных данных (паросочетание):

$$\{(2, 1), (1, 2), (1, 3)\}$$

## 7 Дополнительные требования

Для демонстрации работы программы были приведены дополнительные примеры, в которых рассматриваются разные конфигурации двудольных графов и максимальные паросочетания для них. Эти примеры позволяют оценить работу алгоритма на разных графах с различным количеством вершин и рёбер.