Программная реализация алгоритмов

* Выбранный алгоритм — *нахождение максимального паросочетания*
* Необходимая теория
  + Паросочетанием называется произвольное множество рёбер такое, что никакие два ребра не имеют общей вершины
  + Максимальным парсочетанием называется такое паросочетание *M* в графе *G*, которое не содержится ни в каком другом паросочетании этого графа, то есть к нему невозможно добавить ни одно ребро, которое бы являлось несмежным ко всем рёбрам паросочетания.

Идея алгоритма Эдмондса (Jack Edmonds, 1965 г.) - в **сжатии цветков** (blossom shrinking). Сжатие цветка — это сжатие всего нечётного цикла в одну псевдо-вершину (соответственно, все рёбра, инцидентные вершинам этого цикла, становятся инцидентными псевдо-вершине). Алгоритм Эдмондса ищет в графе все цветки, сжимает их, после чего в графе не остаётся "плохих" циклов нечётной длины, и на таком графе (называемом "поверхностным" (surface) графом) уже можно искать увеличивающую цепь простым обходом в глубину/ширину. После нахождения увеличивающей цепи в поверхностном графе необходимо "развернуть" цветки, восстановив тем самым увеличивающую цепь в исходном графе.Если в графе  существовала увеличивающая цепь, то она существует и в графе , полученном после сжатия цветка, и наоборот.

* Описание алгоритма

1. на ввод подаётся неориентированный граф

2. Найдём нечётный цикл (если такого в данном графе не существует то переходим к пункту 5).

3. Находим наименьшего общего предка.

4. Сожмём этот цикл.

5.Начнём обход в ширину. В процессе обхода в ширину будем строить дерево пути содержащее v, которое будет содержать вершину, которая является паросочетанием для v.

* + - Алгоритм:
    - Начнём обхов в ширину. Находим увеличивающую цепь из свободной вершины root.
    - В процессе обхода рассмотрим текущее ребро (v, to). У нас есть несколько вариантов:
      * Ребро несуществующее. Под этим мы подразумеваем, что v и to принадлежат одной сжатой псевдо-вершине, поэтому в текущем поверхностном графе этого ребра нет. Кроме этого случая, есть ещё один случай: когда ребро (v, to) уже принадлежит текущему паросочетанию; т.к. мы считаем, что вершина v является чётной вершиной, то проход по этому ребру означает в дереве путей подъём к предку вершины v, что недопустимо.
      * Ребро замыкает цикл нечётной длины, т.е. обнаруживается нечётный цикл. В это случае нужно провести сжатие цветка:

1. Для u и v находим наименьшего общего предка (НОП). НОП будет являться псевдо-вершиной

2. Находим те ребра которые находятся в цикле, т.е Проходим от u, v до НОП. Нужно симитировать обход в глубину от псевдо-вершины. Тем самым мы избежим явного объединения списков смежности.

3. Проставим предков для чётных вершин.

* + - Если это «обычное» ребро, то мы делаем обход в ширину. Если мы в вершину to ещё не заходили, и она оказалась ненасыщенной, то мы нашли увеличивающую цепь, заканчивающуюся на вершине to, возвращаем её.
* Инструкция: Запустить программу. Ввести граф в программе. Формат ввода: Сначала нужно ввести количество рёбер в графе. Далее ввести сами ребра u v, где u, v вершины графа. Выводом программы будет изображение формата «.jpg», где красные рёбра будут входить в максимальное паросочетание.
* Асимптотическая сложность. Всего имеется  итераций, на каждой из которых выполняется обход в ширину за , кроме того, могут происходить операции сжатия цветков — их может быть . Сжимаем цветок за O(n). Таким образом, общая асимптотика алгоритма составит .

Скриншоты выполненных тестовых примеров:

Введём такой граф:

9

1 2

2 3

3 4

4 5

6 5

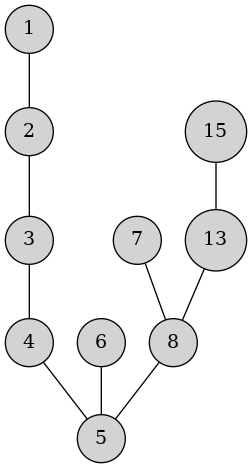
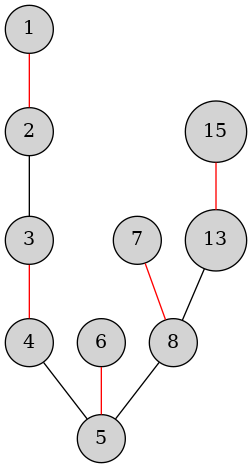
7 8

8 5

13 8

15 13

Изначально граф будет выглядеть так:



8

1 3

1 5

1 6

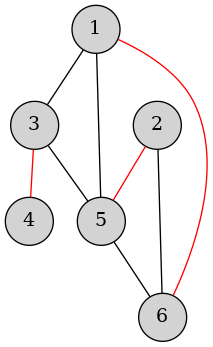
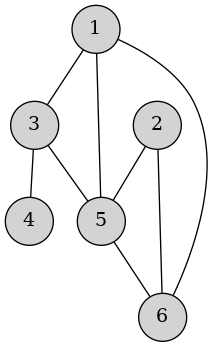
2 5

2 6

3 4

3 5

5 6



6

1 3

15 3

4 1

8 5

5 4

21 15

