Содержание

1 Максимальное паросочетание в двудольном графе.

- Паросочетание.
- Двудольный граф.
- Удлиняющая цепочка это простой путь, который начинается и заканчивается в свободных вершинах, и на котором чередуются ребра из паросочетания и не из паросочетания.
- Паросочетание является максимальным тогда и только тогда, когда в графе нет удлиняющей цепочки.
  - Если есть удлиняющая цепочка, паросочетание не максимальное (очевидно).
  - В обратную сторону докажем от противного. Пусть в M удлиняющей цепочки нет, но это не максимальное паросочетание.
  - Пусть M_{max} — любое максимальное паросочетание.
  - Рассмотрим $M \oplus M_{max}$.
  - Ребра образуют множество путей и циклов.
  - Циклы имеют четную длину.
  - Путь не может начинаться и заканчиваться ребром из M, иначе он бы соответствовал удлиняющей цепочке для M_{max}.
  - Так как $|M_{max}| > |M|$, должен быть хотя бы один путь, который начинается и заканчивается ребром из M_{max}. Он соответствует удлиняющей цепочке для M. Противоречение.
- Алгоритм для нахождения максимального паросочетания в двудольном графе.
  - Поддерживаем паросочетание M.
  - Ребра паросочетания ориентируем справа налево, ребра не из паросочетания — слева направо.
  - Удлиняющая цепочка это любой путь в таком ориентированном графе из свободной вершины левой доли в свободную вершину правой доли.
  - Переберем свободную вершину левой доли, запустим dfs. Если дошли до свободной вершины правой доли, то нашли чередующийся путь.
  - Такой алгоритм работает за $O(n^2 \cdot m)$.

- Можно вместо того, чтобы запускать dfs от каждой из свободных вершин левой доли, запустить один dfs разом от всех них. Тогда алгоритм будет работать за $O(n \cdot m)$.

- Алгоритм Куна.
  - Делаем более-менее то же самое.
  - Первое отличие — dfs будет ходить только по вершинам левой доли. Когда мы идем по ребру (v, u) из левой доли в правую, возможны два варианта:
    - u — свободная. Тогда мы нашли удлиняющую цепочку.
    - u — занята. Тогда из нее есть единственное ребро в левую долю, соответсвующее ребру из паросочетания. Перейдем по нему, и запустим dfs снова из вершины левой доли.
  - Второе отличие — утверждается, что если мы запустили dfs из вершины v левой доли, и не нашли удлиняющий путь, то мы его больше никогда из v и не найдем. Поэтому можно больше из нее dfs не запускать.
  - Таким образом, мы перебираем последовательно все вершины левой доли, пробуем по одному разу из каждой запустить dfs для поиска удлиняющей цепочки.
  - Время работы $O(n \cdot m)$.

- Лемма Холла.
  - Пусть A — какое-то подмножество вершин левой доли. Обозначим за $N(A)$ множество их соседей в правой доле.
  - Полное паросочетание существует тогда и только тогда, когда для любого A верно $|A| \leqslant |N(A)|$.
  - Если есть полное паросочетание, то это очевидно верно.
  - Пусть выполнено это свойство, докажем по индукции, что в графе есть полное паросочетание.
  - База: паросочетание размера 0.
  - Переход: от k к $k+1$. У нас есть паросочетание M ($|M| = k$). Возьмем новую вершину x ($x \notin M$), и запустим из нее dfs. Обозначим за H множество вершин, посещенных dfs-ом.
  - Если $|H_{left}| \leqslant |H_{right}|$, то мы нашли удлиняющую цепочку из x.
  - Если $|H_{left}| > |H_{right}|$, то для множества H_{left} нарушено свойство из леммы Холла.

2 Максимальное паросочетание в произвольном графе.

- Паросочетание максимально тогда и только тогда, когда нет удлиняющей цепочки.
- Попробуем запустить dfs от свободной вершины, который чередует типы ребер.
- Может быть найдем удлиняющую цепочку.
- Может быть не найдем удлиняющую цепочку, хотя она есть. Это означает, что существует вершина, до которой мы могли дойти с разной четность, и дошли с неправильной. Назовем ее сомнительной.
- Если в графе нет сомнительных вершин, и dfs не нашел удлиняющую цепочку, то ее нет. Это следует из того, что если нет сомнительных вершин, то часть графа, достижимая из свободных вершин, является двудольной. А в двудольном графе dfs для поиска удлиняющей цепочки работает корректно.

- Снова попробуем запустить dfs. Он либо найдет удлиняющую цепочку, либо сомнительную вершину, либо ничего.
- Если нашли удлиняющую цепочку, инвертируем ее, и начнем с начала.
- Если не нашли ничего, паросочетание максимально.
- Если нашли сомнительную вершину. Тогда в графе есть соцветие.
- Сожмем цикл соцветия в одну вершину. Все ребра, входившие в цикл, теперь будут входить в новую вершину.
- В исходном графе есть удлиняющая цепочка $\iff$ в графе после сжатия соцветия есть удлиняющая цепочка.

  $\Leftarrow$ Ну там разожмем цикл, цепочка перестраивается.

  $\Rightarrow$
  - Инвертируем стебель соцветия. Это осталось корректное паросочетание. Размер не изменился. Значит оно не максимально. Значит есть удлиняющая цепочка.
  - Сожмем цикл. Конструктивно докажем, что осталась удлиняющая цепочка.
  - Обратно инвертируем стебель. Размер паросочетания не изменился, значит удлиняющая цепочка есть.
- Итого, сжимаем граф, пока нужно. Потом либо не найдем удлиняющую цепочку, тогда конец. Либо найдем. Тогда обратно будем разжимать граф до исходного, там инвертируем удлиняющую цепочку, и начнем с начала. Асимптотика $O(n^2m)$.
- Можно весь процесс сжатия делать в одном запуске dfs-а. Тогда получится $O(nm)$.

3 Потоки и разрезы. Алгоритм Форда-Фалкерсона. Алгоритм Эдмондса-Карпа.

4 Алгоритм Диница. Алгоритм Малхотры-Кумара-Махешвари.

5 Усиленная оценка на время работы алгоритма Диница в специфичных графах. Алгоритм Хопкрофта-Карпа. Алгоритм Push-Relabel.

6 Задача о назначениях. Венгерский алгоритм.

7 Поток минимальной стоимости.