Содержание

1 Расстояния в графе

1.1 Алгоритм Флойда–Уоршелла

Путь $v_{i0} \to v_{i1} \to \cdots \to v_{im}$ длины m называют путём ранга k при $m > 1$, если k – наибольшее из чисел $i_1, \ldots, i_{m-1}$, и путём ранга 0 при $m = 1$. Путь нулевой длины также считают путём ранга 0.
Таким образом, **ранг пути** – максимальный номер вершины, в которую разрешено заходить по пути из v_i в v_j (исключая вершины v_i и v_j).
Обозначим через $C^{(k)}$ матрицу стоимостей прохождения между различными парами вершин по всем путям ранга, не превосходящего k. Элемент $c_{ij}^{(k)}$ содержит стоимость прохождения из вершины v_i в v_j по всем путям рангов $0, 1, \ldots, k-1, k$.
Алгоритм работает на как на неориентированных, так и на ориентированных графах (в этом случае учитывается направление) в том случае, если граф не содержит циклов отрицательного веса.
Идея алгоритма заключается в сравнении стоимостей путей:

- из вершины v_i в вершину v_j по пути ранга не превосходящего $k-1$, т.е. минуя вершину v_k;
- из вершины v_i в вершину v_k, после чего по пути из вершины v_k в вершину v_j.

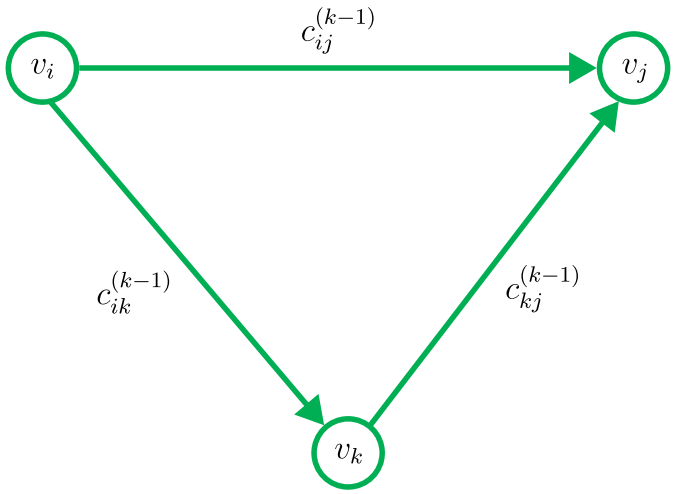


Рис. 1: Граф $G = (V, E)$

Тогда матрицу стоимостей можно найти последовательно вычисляя $C^{(k)}$, $k = \overline{0, k}$ с помощью рекуррентной формулы

$$c_{ij}^{(k)} = \min\left(c_{ij}^{(k-1)}, c_{ik}^{(k-1)} + c_{kj}^{(k-1)}\right).$$

Алгоритм.

1. Инициализируют матрицу стоимостей $C^{(0)}$ с помощью весов рёбер графа.
2. Рекурсивно вычисляют матрицу стоимостей $C^{(k)}$, $k = \overline{0, k}$.
3. Алгоритм заканчивает работу, ранг рассмотренных путей равен числу вершин $k = |V|$.

В случае, если необходимо не только знать кратчайшие расстояния, но и иметь возможность восстановить кратчайший путь, вводится вспомогательная матрица $D^{(k)}$ (см. пример).

Пример.

Дан граф $G = (V, E)$ (рисунок 2).

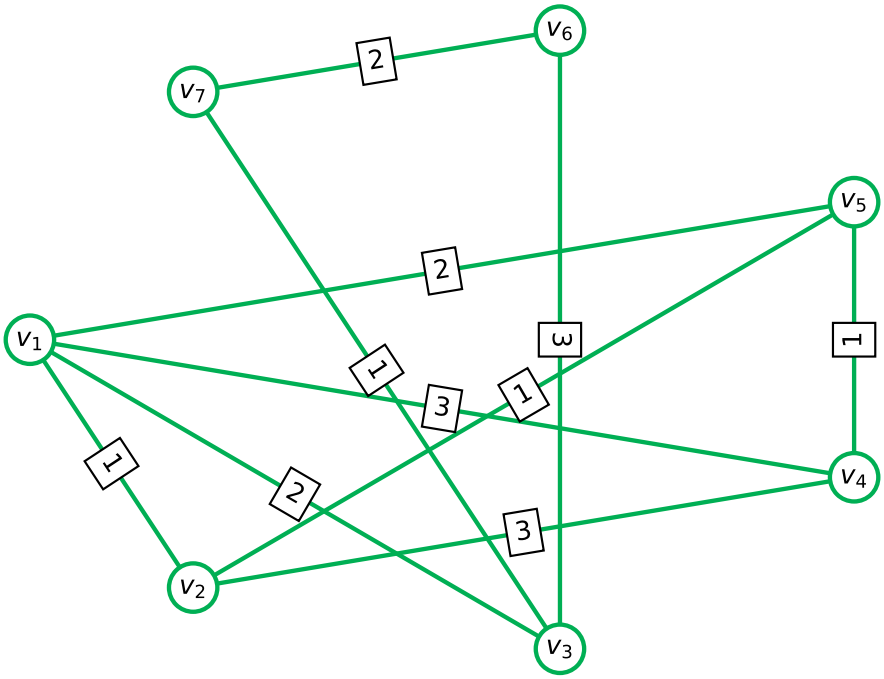


Рис. 2: Граф $G = (V, E)$

При $k = 0$ инициализируем матрицу стоимостей $C^{(0)}$ с помощью весов рёбер.

Для восстановления кратчайших путей введём вспомогательную матрицу $D^{(k)}$ той же размерности, что и $C^{(k)}$. В ячейку $d_{ij}^{(k)}$ на каждой итерации будем помещать значение k, если $c_{ij}^{(k)}$ изменилось.

$$C^{(0)} = \begin{pmatrix} 0 & 1 & 2 & 3 & 2 & \infty & \infty \\ 1 & 0 & \infty & 3 & 1 & \infty & \infty \\ 2 & \infty & 0 & \infty & \infty & 3 & 1 \\ 3 & 3 & \infty & 0 & 1 & \infty & \infty \\ 2 & 1 & \infty & 1 & 0 & \infty & \infty \\ \infty & \infty & 3 & \infty & \infty & 0 & 2 \\ \infty & \infty & 1 & \infty & \infty & 2 & 0 \end{pmatrix} \qquad D^{(0)} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

При $k = 1$ пересчитаем коэффициенты матрицы

$c_{23}^{(1)} = \min(c_{23}^{(0)}, c_{21}^{(0)} + c_{13}^{(0)}) = \min(\infty, 1 + 2) = 3,$

$c_{32}^{(1)} = \min(c_{32}^{(0)}, c_{31}^{(0)} + c_{12}^{(0)}) = \min(\infty, 2 + 1) = 3,$

$c_{34}^{(1)} = \min(c_{34}^{(0)}, c_{31}^{(0)} + c_{14}^{(0)}) = \min(\infty, 2 + 3) = 5,$

$c_{43}^{(1)} = \min(c_{43}^{(0)}, c_{41}^{(0)} + c_{13}^{(0)}) = \min(\infty, 3 + 2) = 5,$

$c_{35}^{(1)} = \min(c_{35}^{(0)}, c_{31}^{(0)} + c_{15}^{(0)}) = \min(\infty, 2 + 2) = 4,$

$c_{53}^{(1)} = \min(c_{53}^{(0)}, c_{51}^{(0)} + c_{13}^{(0)}) = \min(\infty, 2 + 2) = 4,$

$$C^{(1)} = \begin{pmatrix} 0 & 1 & 2 & 3 & 2 & \infty & \infty \\ 1 & 0 & 3 & 3 & 1 & \infty & \infty \\ 2 & 3 & 0 & 5 & 4 & 3 & 1 \\ 3 & 3 & 5 & 0 & 1 & \infty & \infty \\ 2 & 1 & 4 & 1 & 0 & \infty & \infty \\ \infty & \infty & 3 & \infty & \infty & 0 & 2 \\ \infty & \infty & 1 & \infty & \infty & 2 & 0 \end{pmatrix} \qquad D^{(1)} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

При $k = 2$ коэффициенты матрицы не меняются

$$C^{(2)} = \begin{pmatrix} 0 & 1 & 2 & 3 & 2 & \infty & \infty \\ 1 & 0 & 3 & 3 & 1 & \infty & \infty \\ 2 & 3 & 0 & 5 & 4 & 3 & 1 \\ 3 & 3 & 5 & 0 & 1 & \infty & \infty \\ 2 & 1 & 4 & 1 & 0 & \infty & \infty \\ \infty & \infty & 3 & \infty & \infty & 0 & 2 \\ \infty & \infty & 1 & \infty & \infty & 2 & 0 \end{pmatrix} \qquad D^{(2)} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

При $k = 3$ пересчитаем коэффициенты матрицы

$c_{16}^{(3)} = \min(c_{16}^{(2)}, c_{13}^{(2)} + c_{36}^{(2)}) = \min(\infty, 2 + 3) = 5,$
$c_{61}^{(3)} = \min(c_{61}^{(2)}, c_{63}^{(2)} + c_{31}^{(2)}) = \min(\infty, 3 + 2) = 5,$
$c_{17}^{(3)} = \min(c_{17}^{(2)}, c_{13}^{(2)} + c_{37}^{(2)}) = \min(\infty, 2 + 1) = 3,$
$c_{71}^{(3)} = \min(c_{71}^{(2)}, c_{73}^{(2)} + c_{31}^{(2)}) = \min(\infty, 1 + 2) = 3,$
$c_{46}^{(3)} = \min(c_{46}^{(2)}, c_{43}^{(2)} + c_{36}^{(2)}) = \min(\infty, 5 + 3) = 8,$
$c_{64}^{(3)} = \min(c_{64}^{(2)}, c_{63}^{(2)} + c_{34}^{(2)}) = \min(\infty, 3 + 5) = 8,$
$c_{47}^{(3)} = \min(c_{47}^{(2)}, c_{43}^{(2)} + c_{37}^{(2)}) = \min(\infty, 5 + 1) = 6,$
$c_{74}^{(3)} = \min(c_{74}^{(2)}, c_{73}^{(2)} + c_{34}^{(2)}) = \min(\infty, 1 + 5) = 6,$
$c_{56}^{(3)} = \min(c_{56}^{(2)}, c_{53}^{(2)} + c_{36}^{(2)}) = \min(\infty, 4 + 3) = 7,$
$c_{65}^{(3)} = \min(c_{65}^{(2)}, c_{63}^{(2)} + c_{35}^{(2)}) = \min(\infty, 3 + 4) = 7,$
$c_{57}^{(3)} = \min(c_{57}^{(2)}, c_{53}^{(2)} + c_{37}^{(2)}) = \min(\infty, 4 + 1) = 5,$
$c_{75}^{(3)} = \min(c_{75}^{(2)}, c_{73}^{(2)} + c_{35}^{(2)}) = \min(\infty, 1 + 4) = 5,$

$$C^{(3)} = \begin{pmatrix} 0 & 1 & 2 & 3 & 2 & 5 & 3 \\ 1 & 0 & 3 & 3 & 1 & 6 & 4 \\ 2 & 3 & 0 & 5 & 4 & 3 & 1 \\ 3 & 3 & 5 & 0 & 1 & 8 & 6 \\ 2 & 1 & 4 & 1 & 0 & 7 & 5 \\ 5 & 6 & 3 & 8 & 7 & 0 & 2 \\ 3 & 4 & 1 & 6 & 5 & 2 & 0 \end{pmatrix} \qquad D^{(3)} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 3 & 3 \\ 0 & 0 & 1 & 0 & 0 & 3 & 3 \\ 3 & 0 & 0 & 3 & 3 & 0 & 0 \\ 3 & 0 & 0 & 3 & 3 & 0 & 0 \end{pmatrix}.$$

При $k = 4$ коэффициенты матрицы не меняются

$$C^{(4)} = \begin{pmatrix} 0 & 1 & 2 & 3 & 2 & 5 & 3 \\ 1 & 0 & 3 & 3 & 1 & 6 & 4 \\ 2 & 3 & 0 & 5 & 4 & 3 & 1 \\ 3 & 3 & 5 & 0 & 1 & 8 & 6 \\ 2 & 1 & 4 & 1 & 0 & 7 & 5 \\ 5 & 6 & 3 & 8 & 7 & 0 & 2 \\ 3 & 4 & 1 & 6 & 5 & 2 & 0 \end{pmatrix} \qquad D^{(4)} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 3 & 3 \\ 0 & 0 & 1 & 0 & 0 & 3 & 3 \\ 3 & 0 & 0 & 3 & 3 & 0 & 0 \\ 3 & 0 & 0 & 3 & 3 & 0 & 0 \end{pmatrix}.$$

При $k = 5$ пересчитаем коэффициенты матрицы
$c_{24}^{(5)} = \min(c_{24}^{(4)}, c_{25}^{(4)} + c_{54}^{(4)}) = \min(3, 1+1) = 2,$
$c_{42}^{(5)} = \min(c_{42}^{(4)}, c_{45}^{(4)} + c_{52}^{(4)}) = \min(3, 1+1) = 2,$

$$C^{(5)} = \begin{pmatrix} 0 & 1 & 2 & 3 & 2 & 5 & 3 \\ 1 & 0 & 3 & 2 & 1 & 6 & 4 \\ 2 & 3 & 0 & 5 & 4 & 3 & 1 \\ 3 & 2 & 5 & 0 & 1 & 8 & 6 \\ 2 & 1 & 4 & 1 & 0 & 7 & 5 \\ 5 & 6 & 3 & 8 & 7 & 0 & 2 \\ 3 & 4 & 1 & 6 & 5 & 2 & 0 \end{pmatrix} \qquad D^{(5)} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 1 & 5 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 5 & 1 & 0 & 0 & 3 & 3 \\ 0 & 0 & 1 & 0 & 0 & 3 & 3 \\ 3 & 0 & 0 & 3 & 3 & 0 & 0 \\ 3 & 0 & 0 & 3 & 3 & 0 & 0 \end{pmatrix}.$$

При $k = 6$ коэффициенты матрицы не меняются

$$C^{(6)} = \begin{pmatrix} 0 & 1 & 2 & 3 & 2 & 5 & 3 \\ 1 & 0 & 3 & 2 & 1 & 6 & 4 \\ 2 & 3 & 0 & 5 & 4 & 3 & 1 \\ 3 & 2 & 5 & 0 & 1 & 8 & 6 \\ 2 & 1 & 4 & 1 & 0 & 7 & 5 \\ 5 & 6 & 3 & 8 & 7 & 0 & 2 \\ 3 & 4 & 1 & 6 & 5 & 2 & 0 \end{pmatrix} \qquad D^{(6)} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 1 & 5 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 5 & 1 & 0 & 0 & 3 & 3 \\ 0 & 0 & 1 & 0 & 0 & 3 & 3 \\ 3 & 0 & 0 & 3 & 3 & 0 & 0 \\ 3 & 0 & 0 & 3 & 3 & 0 & 0 \end{pmatrix}.$$

При $k = 7$ коэффициенты матрицы не меняются

$$C^{(7)} = \begin{pmatrix} 0 & 1 & 2 & 3 & 2 & 5 & 3 \\ 1 & 0 & 3 & 2 & 1 & 6 & 4 \\ 2 & 3 & 0 & 5 & 4 & 3 & 1 \\ 3 & 2 & 5 & 0 & 1 & 8 & 6 \\ 2 & 1 & 4 & 1 & 0 & 7 & 5 \\ 5 & 6 & 3 & 8 & 7 & 0 & 2 \\ 3 & 4 & 1 & 6 & 5 & 2 & 0 \end{pmatrix} \qquad D^{(7)} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 1 & 5 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 5 & 1 & 0 & 0 & 3 & 3 \\ 0 & 0 & 1 & 0 & 0 & 3 & 3 \\ 3 & 0 & 0 & 3 & 3 & 0 & 0 \\ 3 & 0 & 0 & 3 & 3 & 0 & 0 \end{pmatrix}.$$

Итоговая матрица кратчайших расстояний C и вспомогательная матрица D имеют вид

$$C = \begin{pmatrix} 0 & 1 & 2 & 3 & 2 & 5 & 3 \\ 1 & 0 & 3 & 2 & 1 & 6 & 4 \\ 2 & 3 & 0 & 5 & 4 & 3 & 1 \\ 3 & 2 & 5 & 0 & 1 & 8 & 6 \\ 2 & 1 & 4 & 1 & 0 & 7 & 5 \\ 5 & 6 & 3 & 8 & 7 & 0 & 2 \\ 3 & 4 & 1 & 6 & 5 & 2 & 0 \end{pmatrix} \qquad D = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 1 & 5 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 5 & 1 & 0 & 0 & 3 & 3 \\ 0 & 0 & 1 & 0 & 0 & 3 & 3 \\ 3 & 0 & 0 & 3 & 3 & 0 & 0 \\ 3 & 0 & 0 & 3 & 3 & 0 & 0 \end{pmatrix}.$$

Восстановим путь из v_4 в v_6

$$v_4 \to \cdots \to v_6,$$

так как $d_{46} = 3 \neq 0$, добавляем в путь вершину v_3,

$$v_4 \to \cdots \to v_3 \to v_6,$$

так как $d_{43} = 1 \neq 0$, добавляем в путь вершину v_1, а так как $d_{41} = 0$, восстановление закончено, полученный путь

$$v_4 \to v_1 \to v_3 \to v_6.$$

2 Деревья

2.1 Определения

Деревом (неориентированным) называют связный ациклический граф.

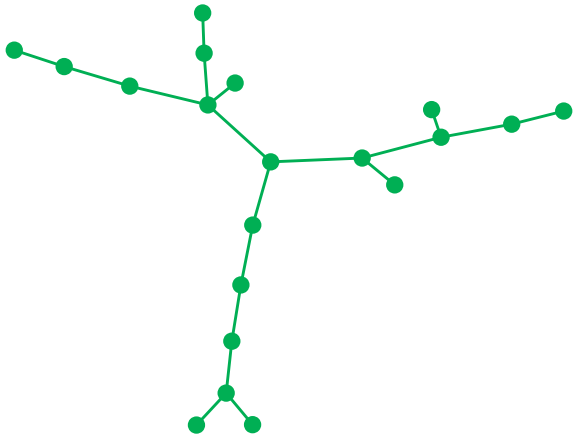


Рис. 3: Пример неориентированного дерева

Ориентированным деревом называют бесконтурный ориентированный граф, у которого полустепень захода любой вершины не больше 1 и существует ровно одна вершина, называемая **корнем ориентированного дерева**, полустепень захода которой равна 0.
Вершину v ориентированного дерева называют **потомком** вершины u, если существует путь из v в u. В этом же случае вершину u называют **предком** вершины v.
Вершину ориентированного дерева, не имеющую потомков называют **листом**.
Высота ориентированного дерева — это наибольшая длина пути из корня в лист.
Глубина вершины ориентированного дерева — это длина пути из корня в эту вершину.
Высота вершины -– это наибольшая длина пути из данной вершины в лист.
Уровень вершины ориентированного дерева — это разность между высотой ориентированного дерева и глубиной данной вершины.
Лес – ациклический граф, состоящий из нескольких компонент связности.
Для деревьев справедливы следующие утверждения:

1. $|V| = |E| + 1$;
2. Если добавить к дереву любое ребро, не добавляя вершин, в графе возникнет цикл.
3. Если удалить любое ребро, не удаляя вершин, граф перестанет быть связным.

2.2 Минимальное остовное дерево. Алгоритм Краскала

Остовным деревом графа $G = (V, E)$ называют дерево, являющееся подграфом $F \subseteq G$, такое что $F = (V, E')$.
Взешенным называется граф, каждому ребру которого поставлено в соответствие некоторое число – вес, иначе говоря задана функция $f : E \to \mathbb{R}$.

Пусть необходимо отыскать остовное дерево, имеющее наименьший суммарный вес рёбер.

Алгоритм Краскала.

Дан неориентированный взвешенной связный граф $G = (V, E)$, при этом $n = |V|$, $m = |E|$.

Для поиска минимального остовного дерева на каждом шаге будем так выбирать ребро минимального веса, чтобы не возникало циклов.

Алгоритм заканчивает работу когда выбрано $n - 1$ ребро.

1. Упорядочить рёбра по возрастанию весов: $e_1, e_2, \ldots, e_m$, $f(e_1) \leq \cdots \leq f(e_m)$.
2. Выбрать первое ребро и занести его в класс K_1.
3. Выбрать следующее ребро $e = \{u, v\}$. Возможны 4 случая:
   - Ни u, ни v не инцидентны никаким ребрам из имеющихся классов $K_1, K_2, \ldots, K_i$. Создаётся новый класс K_{i+1}.
   - Одна вершина v не инцидентна никакому ребру из $K_1, K_2, \ldots, K_i$, а вершина u инцидентна некоторому ребру из K_j. Ребро $\{u, v\}$ добавляется в класс K_j.
   - Вершины инцидентны ребрам из разных классов K_j и K_l . Классы объединяются в один, содержащий также ребро $\{u, v\}$.
   - Обе вершины инцидентны ребрам из одного класса. Ребро не добавляется в дерево (иначе образуется цикл).

   Шаг 3 повторяется $n - 2$ раза.

Покажем работу алгоритма на примере графа с рисунка 4.

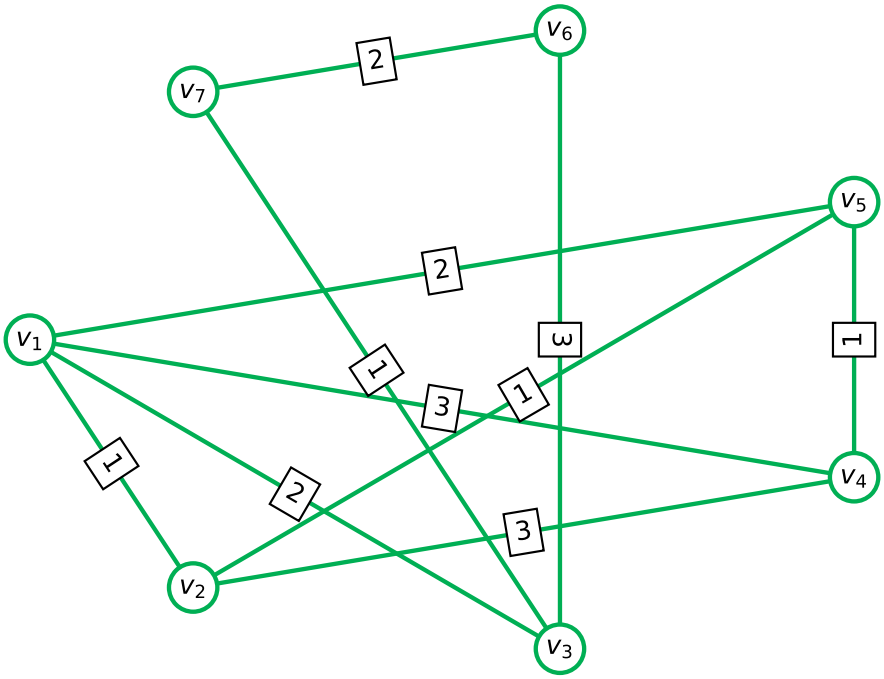


Рис. 4: Взвешенный связный граф

Отсортируем рёбра по увеличению веса.

Вес	Ребро
1	$\{v_1, v_2\}$
1	$\{v_1, v_3\}$
1	$\{v_2, v_5\}$
1	$\{v_3, v_7\}$
1	$\{v_4, v_5\}$
2	$\{v_1, v_5\}$
2	$\{v_6, v_7\}$
3	$\{v_1, v_4\}$
3	$\{v_2, v_4\}$
3	$\{v_3, v_6\}$

Покажем основные этапы на рисунке 5.

a) $K_1 = \{\{v_1, v_2\}\}$.

b) $K_1 = \{\{v_1, v_2\}, \{v_1, v_3\}\}$.

c) $K_1 = \{\{v_1, v_2\}, \{v_1, v_3\}, \{v_2, v_5\}\}$.

d) $K_1 = \{\{v_1, v_2\}, \{v_1, v_3\}, \{v_2, v_5\}, \{v_3, v_7\}\}$.

e) $K_1 = \{\{v_1, v_2\}, \{v_1, v_3\}, \{v_2, v_5\}, \{v_3, v_7\}, \{v_4, v_5\}\}$.

f) Следующее по очереди ребро $\{v_1, v_5\}$, но при добавлении его получим цикл на вершинах $v_1 - v_2 - v_5 - v_1$, поэтому вместо него берём ребро $\{v_6, v_7\}$.

$K_1 = \{\{v_1, v_2\}, \{v_1, v_3\}, \{v_2, v_5\}, \{v_3, v_7\}, \{v_4, v_5\}, \{v_6, v_7\}\}$.

В нашем графе $n = |V| = 7$, сделано $n - 1 = 6$ шагов, работа алгоритма завершена.

Минимальное остовное дерево показано на рисунке 6.

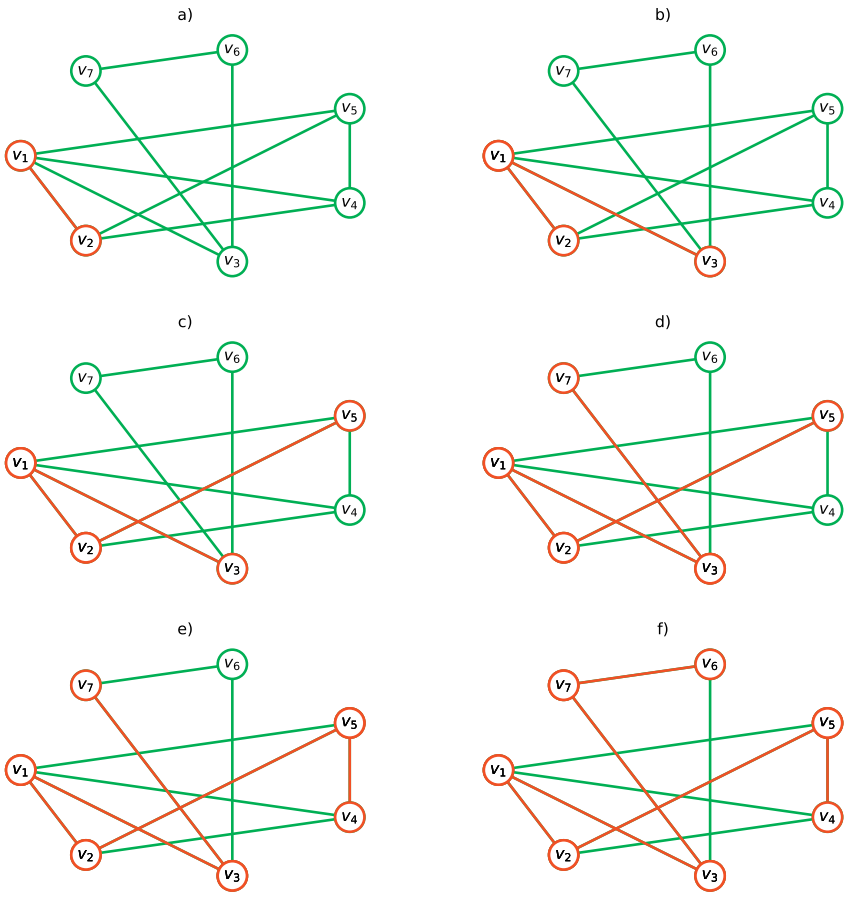


Рис. 5: Шаги алгоритма Краскала

3 Циклы

3.1 Циклы Эйлера

Связный граф называется **эйлеровым**, если можно обойти все рёбра ровно по одному разу и вернуться в исходную вершину.

Лемма 3.1. *Пусть в графе степень каждой вершины не меньше 2, тогда граф содержит цикл.*

Доказательство. Пусть $v_1 \in V$, тогда построим последовательность рёбер $\{v_1, v_2\}, \{v_2, v_3\}, \ldots$ выбирая v_2 смежной с v_1 и отличной от v_1. По условию теоремы на каждом шаге очередная вершина существует. В силу конечности множества вершин V в последовательности встретится вершина v_k, которая встречалась раньше. Выбирая k минимальным с этим свойством, получим цикл. □

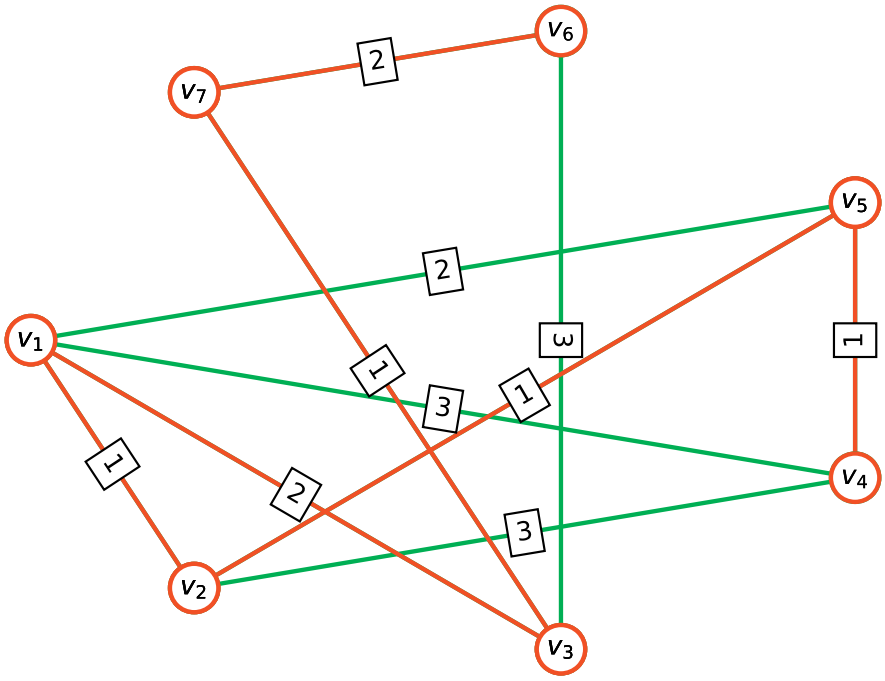
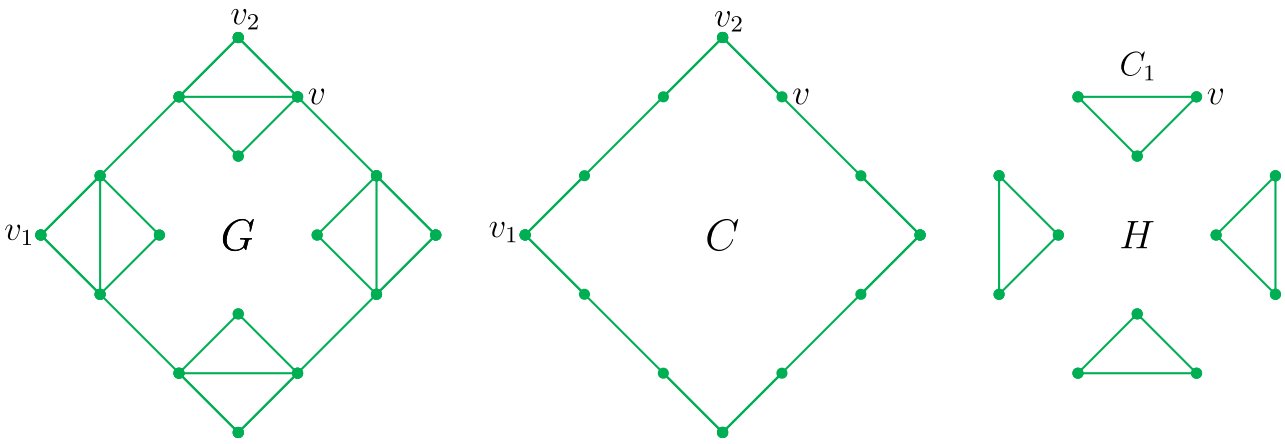


Рис. 6: Минимальное остовное дерево

Теорема 3.2. *Связный граф является эйлеровым тогда и только тогда, когда степени всех его вершин четны.*

Доказательство. Предположим, что P является эйлеровым циклом в графе G. Тогда при всяком прохождении цикла через любую вершину графа используется одно ребро для входа и одно ребро для выхода. Поскольку каждое ребро используется один раз, то каждая вершина должна иметь четную степень.
Достаточность доказывается индукцией по числу рёбер n.
База индукции: $n = 0$ цикл существует.
Пусть граф, имеющий менее n рёбер содержит эйлеров цикл.
Рассмотрим связный граф $G = (V, E)$ с $n > 0$ рёбрами, все степени вершин которого чётны.



Пусть v_1 и v_2 – вершины графа. В силу связности G, существует путь из v_1 в v_2.
$\deg(v_2)$ – чётная, значит существует неиспользованное ребро, по которому можно продолжить путь из v_2.
Так как граф конечный, то путь должен вернуться в v_1, что образует цикл C. Если есть рёбра, инцидентные v_1, но не включённые в C, продолжаем строить C через v_1 таким же образом.
Если C является эйлеровым циклом для G, тогда доказательство закончено.
Если нет, то пусть H – подграф графа G, полученный удалением всех рёбер, принадлежащих C. Поскольку C содержит чётное число рёбер, инцидентных каждой вершине, то каждая вершина подграфа H имеет чётную степень. А так как C покрывает все ребра, инцидентные v_1, то граф H будет состоять из нескольких компонент связности.
Поскольку каждая компонента связности H имеет менее, чем n рёбер, а у каждой вершины графа H чётная степень, то у каждой компоненты связности H существует эйлеров цикл.
Рассмотрим какую-либо компоненту связности с эйлеровым циклом C_1. У C и C_1 имеется общая вершина v, в силу связности G. Теперь можно обойти эйлеров цикл, начиная его в вершине v, обойти C , вернуться в v, затем пройти C_1 и вернуться в v. Если новый эйлеров цикл не является эйлеровым циклом для G, продолжаем данное построение эйлерового цикла, пока не получим эйлеров цикл для G.

□

Если в графе ровно две вершины имеют нечётную степень, он называется **полуэйлеровым**.
Ребро, при удалении которого граф теряет связность, называется **мостом**.

3.2 Алгоритм Флёри

Алгоритм используется для поиска эйлерова цикла в графе.

1. Если граф эйлеров, начинаем движение из любой вершины, если полуэйлеров, то из одной из нечётных.

2. Из каждой вершины проходят по любому инцидентному ребру, только если оно не является мостом, после чего пройденные рёбра удаляют из графа.

3. Алгоритм заканчивает работу, когда рёбер в графе больше нет.

4 Фундаментальные системы циклов и разрезов

Объединением графов $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$ будем называть объединение множеств вершин и рёбер $G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2)$.
Цикл в графе – подграф, изоморфный C_n.
Обозначим цикл через последовательность входящих в него вершин

$$c = (v_1, v_2, \ldots, v_k),$$

где $\{v_i, v_{i+1}\} \in E, i = \overline{1, k-1}, \{v_k, v_1\} \in E$.
Полицикл в графе – объединение нескольких циклов графа, не имеющих общих рёбер.
Например, в графе на рисунке 7 $(v_1, v_2, v_5) \cup (v_2, v_3, v_4)$ – полицикл.
Введём операцию сложения по модулю два $\oplus$ для циклов фиксированного графа.

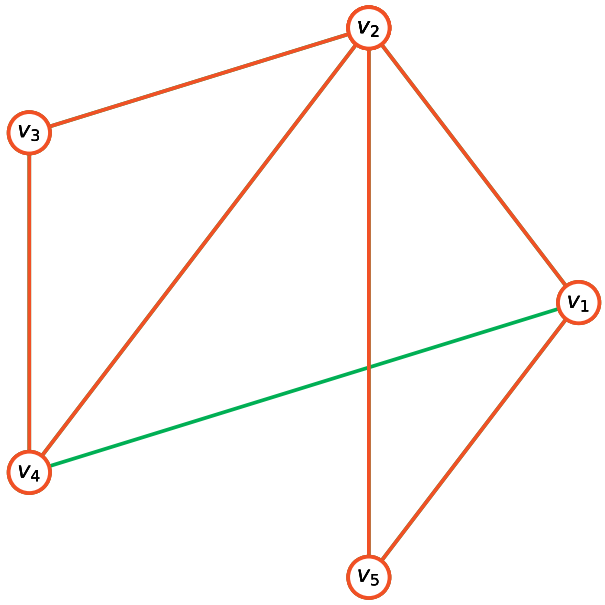


Рис. 7: Полицикл выделен красным цветом

Подграф $c_1 \oplus c_2$ – объединение циклов c_1 и c_2, при котором их общие рёбра удаляются из объединения. В теории множеств аналогом сложению по модулю 2 является симметрическая разность.
Например, $(v_1, v_2, v_4) \oplus (v_2, v_3, v_4) = (v_1, v_2, v_3, v_4)$, общее ребро $\{v_2, v_4\}$ удалено.
Обозначим множество всех полициклов графа P. Считаем, что в P входит и пустой цикл – пустое множество.
Множество полициклов P в любом графе с операцией сложения по модулю два $\oplus$ и умножением на 0 и 1 является линейным пространством.

4.1 Фундаментальная система циклов

Фундаментальная система циклов (ФСЦ) – базис этого пространства P, то есть такая минимальная совокупность циклов, линейной комбинацией которых можно представить любой цикл графа.
Для нахождения фундаментальной системы циклов, ассоциированной с заданным остовным деревом T, необходимо все рёбра графа разделить на древесные $d_1, d_2, \ldots, d_m$ и недревесные $e_1, e_2, \ldots, e_k$.
При добавлении любого недревесного ребра e_i к T образуется цикл c_i. Циклы $c_1, \ldots, c_k$ и образуют фундаментальную систему.
Для того чтобы выразить любой цикл графа, нужно сложить те циклы фундаментальной системы, которые соответствуют недревесным рёбрам раскладываемого цикла.
На рисунке 8 показано остовное дерево (a) и три цикла (b,c,d), образующих ФСЦ.
Для обоснования корректности алгоритма нужно показать, что:

1. При добавлении ребра к остову образуется ровно один цикл.

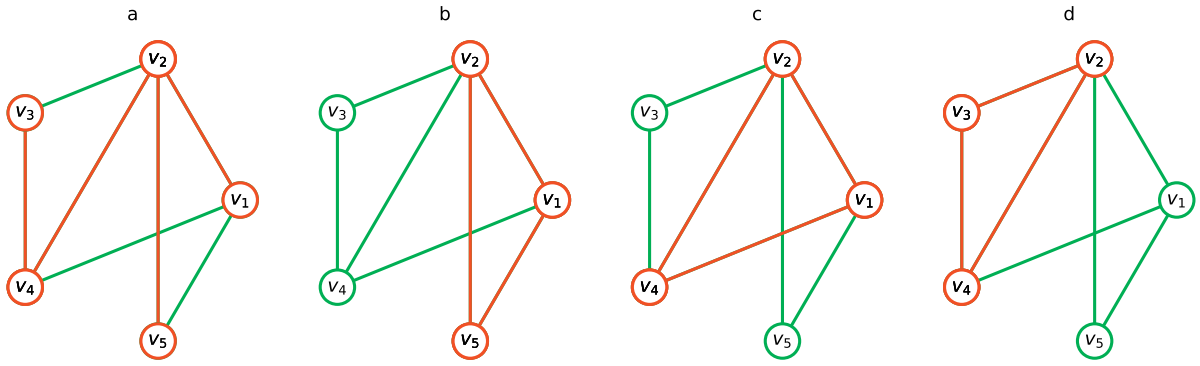


Рис. 8: Фундаментальная система циклов

2. При сложении циклов фундаментальной системы, соответствующих рёбрам цикла, получится искомый цикл.

Если два цикла c_1, c_2 содержат общее ребро, то имеется полицикл $c_1 \oplus c_2$, не содержащий этого ребра. Так как в дереве полицикла не может быть по определению, пункт 1 доказан.
Для доказательства пункта 2 покажем сначала, что при сложении циклов останутся все недревесные рёбра. Действительно, каждое недревесное ребро содержится в своём цикле. Значит, оно не может сократиться из-за операции сложения.
Предположим, что в графе существует два цикла с одинаковым набором недревесных рёбер, тогда $c_1 \oplus c_2$ есть полицикл, не содержащий недревесных рёбер. Противоречие, а значит существует лишь один цикл с заданным наором недревесных рёбер.
Корректность алгоритма доказана.

4.2 Фундаментальная система разрезов

Разрезом в графе называется минимальное по включению множество рёбер, при удалении которого граф становится несвязным.
На множестве разрезов можно ввести ту же операцию $\oplus$, что и для циклов.
Фундаментальная система разрезов (ФСР) – минимальная по включению система разрезов, через которую можно выразить любой разрез графа.
Для нахождения фундаментальной системы разрезов (ФСР), ассоциированной с заданным остовом графа, необходимо найти все фундаментальные разрезы.
Для нахождения фундаментального разреза, убираем одно ребро остовного дерева, при этом в остове образуются две новые компоненты связности. Множество рёбер графа, соединяющих эти компоненты остова, вместе с удалённым ребром остова образуют разрез.
Проделав эту процедуру со всеми рёбрами, входящими в остов, получаем систему разрезов. Она является фундаментальной.
Для того чтобы выразить произвольный разрез через фундаментальную систему, нужно сложить разрезы, соответствующие древесным рёбрам данного разреза.
На рисунке 9 показано остовное дерево (a) и четыре разреза (b,c,d,e), образующих ФСЦ.

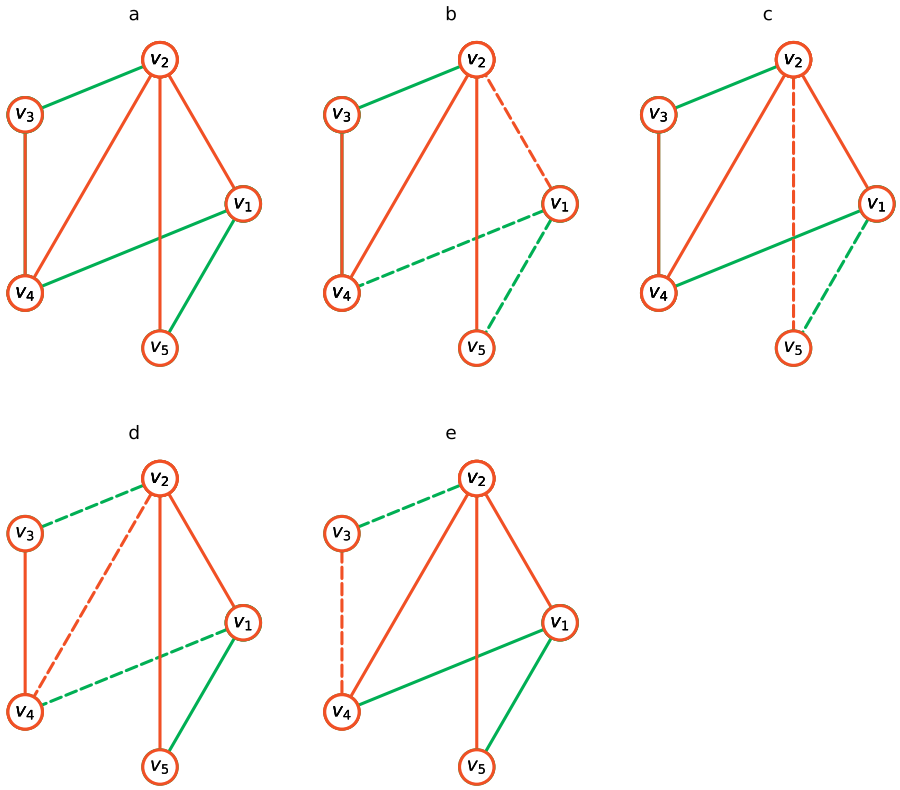


Рис. 9: Фундаментальная система разрезов