Содержание

1 Матрица достижимости

1.1 Композиция бинарных отношений

Пусть на множествах A, B и C определены бинарные отношения

$$R \subseteq A \times B,$$

$$Q \subseteq B \times C.$$

Тогда композицией отношений R и Q называется такое отношение $(R \circ Q)$, что

$$\forall a \in A, c \in C : a(R \circ Q)c \Leftrightarrow \exists b \in B : (aRb) \wedge (bQc).$$

Пример
$A = \{a, b, c, d\}, B = \{1, 2, 3, 4\}, C = \{!, ?, *, \backslash\}$.
$R = \{(a, 2), (b, 2), (c, 3), (c, 4)\}, Q = \{(1, !), (2, *), (4, \backslash)\}$.

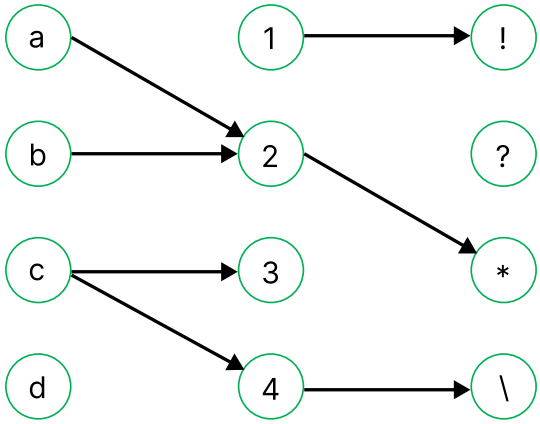

Рис. 1: Бинарные отношения R и Q

Матрицы бинарных отношений имеют вид

$$M_R = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad M_Q = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Рассмотрим композицию отношений $R \circ Q = \{(a, *), (b, *), (c, \backslash)\}$.
Матрицу композиций можно определить с помощью булева произведения матриц M_R и M_Q

$$M_{R \circ Q} = M_R \cdot M_Q = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

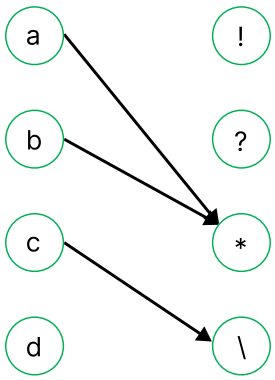

Рис. 2: Композиция отношений

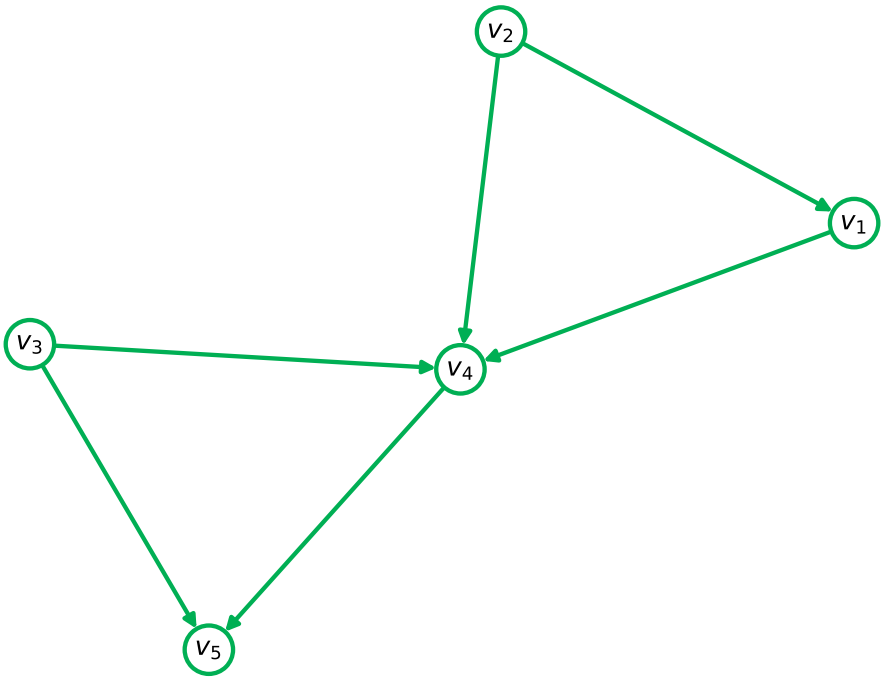

Рис. 3: $G = (V, E)$

1.2 Матрица достижимости

Рассмотрим ориентированный граф $G = (V, E)$ на рисунке 3.
Его матрица смежности имеет вид

$$S(G) = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Элемент $s_{ij} = 1$ в том случае, если вершины v_1 и v_j смежны, а значит существует путь длины 1 из v_1 в v_j.

Для поиска путей длины 2 можно рассмотреть композицию отношения смежности с собой, матрица такого отношения будет определяться как

$$S^2 = S \cdot S = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

.

Для поиска путей длины 3 можно воспользоваться матрицей $S^3 = S^2 \cdot S$.

$$S^3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Мы можем определить матрицу достижимости $R(G)$ как матрицу рефлексивно-транзитивного замыкания отношения смежности, тогда

$$R(G) = E \vee S \vee S^2 \vee \cdots \vee S^{n-1}.$$

Для нашего графа S^4=0, поэтому

$$R(G) = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

2 Задачи об обходах графа

Для определения количества и состава компонент связности используют алгоритмы обхода графов: поиск в глубину и поиск в ширину.
Рассмотрим алгоритмы обхода на примере графа $G = (V, E)$ с рисунка 4.

2.1 Поиск в глубину (англ. Depth-first search, DFS)

Идея DFS состоит в том, чтобы идти «вглубь» графа, насколько это возможно.
Алгоритм поиска описывается рекурсивно: перебираем все исходящие из рассматриваемой вершины рёбра. Если ребро ведёт в вершину, которая не была посещена ранее, то запускаем алгоритм от этой непосещённой вершины, а после возвращаемся и продолжаем перебирать рёбра. Возврат происходит в том случае, если не осталось смежных с рассматриваемой вершиной непосещённых вершин.
Если после завершения алгоритма не все вершины были рассмотрены, то необходимо запустить алгоритм от одной из нерассмотренных вершин.
Описание алгоритма.

1. Пронумеруем все вершины.

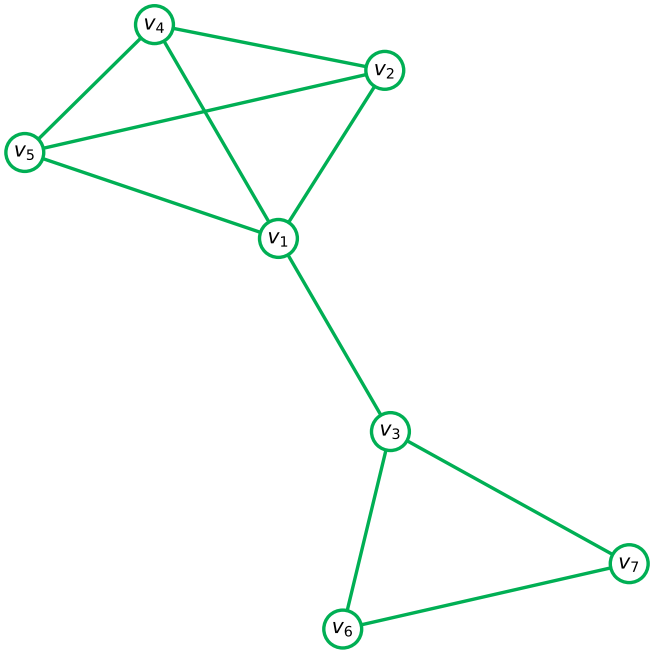

Рис. 4: Граф $G = (V, E)$

2. Выбираем вершину с наименьшим номером и отмечаем как посещённую.

3. Выбираем смежную к текущей вершину с наименьшим номером и помечаем её посещённой, она станет текущей на следующем шаге. Если таких вершин нет, возвращаемся к предыдущей посещённой вершине.

4. Повторяем шаг 3, пока не произойдёт возврат в начальную вершину.

5. Если после этого остались непройденные вершины, значит граф несвязный. В этом случае берём непройденную вершину с наименьшим номером, из которой производим обход следующей компоненты связности.

Шаги алгоритма показаны на рисунке 5.

2.2 Поиск в ширину (англ. Breadth-first search, BFS)

Идея алгоритма состоит в распределении вершин по уровням, характеризующим удалённость от первой вершины.

Описание алгоритма.

1. Возьмём вершину v_1, как вершину нулевого уровня, отмечаем её как пройденную.

2. Формируем уровень i+1 из непройденных вершин, смежных с вершинами на i-го уровня.

3. Если множество вершин i+1 уровня не пусто, помечаем их как пройденные и формируем следующий уровень.

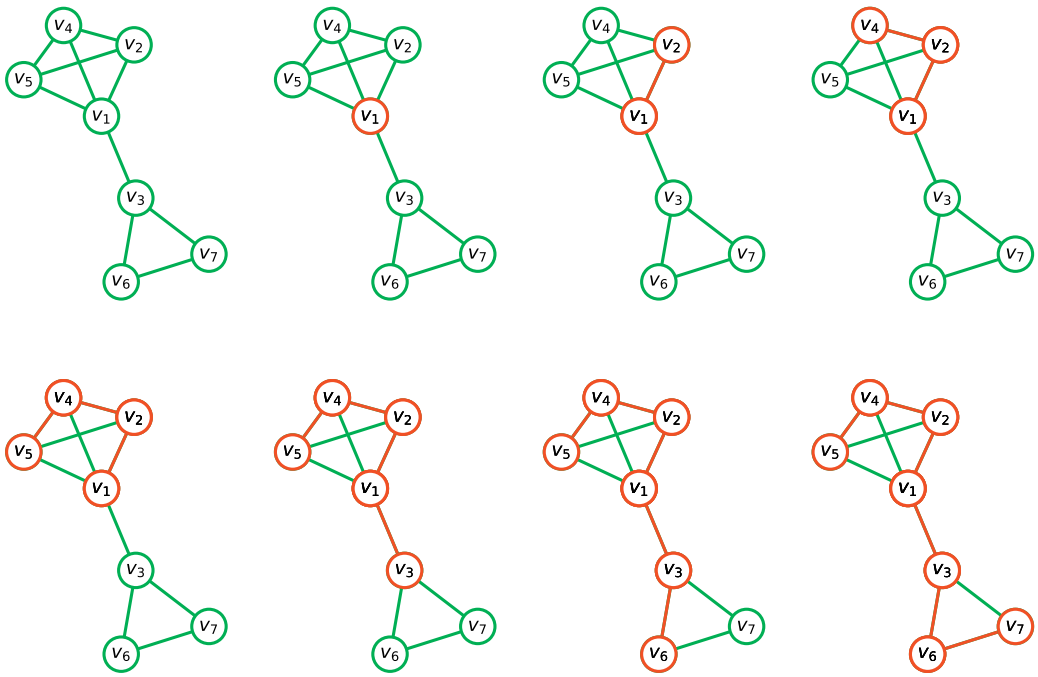

Рис. 5: Поиск в глубину

4. Если уровень оказывается пустым, обход компоненты связности завершён.
5. Если пройдены не все вершины, выполняют обход других компонент.

Шаги алгоритма показаны на рисунке 6.

2.3 Алгоритм Косарайю для поиска компонент сильной связности

Описание алгоритма

1. Запускаем DFS на исходном графе G, запоминая времена выхода $\tau(v)$ для каждой вершины.
2. Транспонируем граф $G^T = (G)^T$, меняя направление дуг на противоположное.
3. Запускаем DFS на транспонированном графе G^T, выбирая в качестве стартовой непосещённую вершину с максимальным значением $\tau(v)$.
4. Полученные в шаге 3 подграфы и образуют компоненты сильной связности.

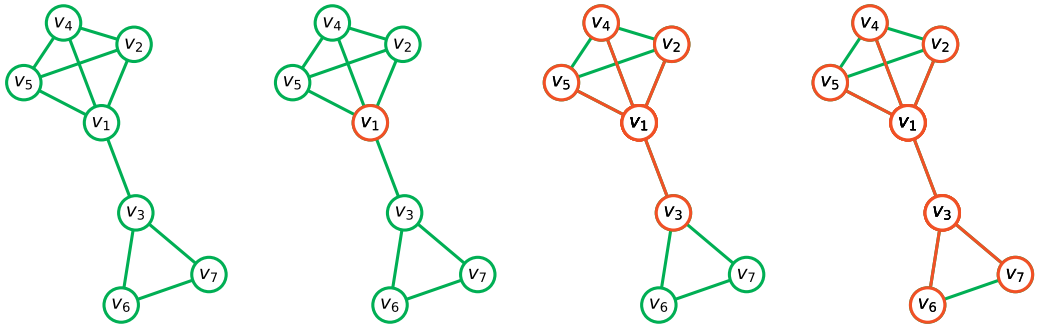

Рис. 6: Поиск в ширину

2.4 Конденсация и база графа

Введём обозначение $R(v)$ для множества вершин, достижимых из вершины v.
Множество вершин $\{v_1, v_2, \ldots, v_k\} \subseteq V$ ориентированного графа $G = (V, E)$ называется его **базой**, если из вершин этого множества достижимы все остальные вершины и оно минимально по включению.
База существует в любом ориентированном графе. Любые две вершины базы не достижимы одна из другой.
Вершины с нулевыми степенями захода обязаны принадлежать базе.
Для построения базы используется конденсация графа, то есть такой граф G^c, в котором каждая сильно связная компонента исходного графа стянута в одну вершину.
Граф конденсации $G^c = (V^c, E^c)$ ориентированного графа $G = (V, E)$ – ориентированный граф с множеством вершин

$$V^c = \{V_1, V_2, \ldots, V_k\},$$

где каждая V_i – множество вершин сильно связной компоненты орграфа G.
Вершины V_i и V_j графа конденсаций соединяются дугой, если в исходном ориентированном графе существовала дуга (v, u), где $v \in V_i$, $u \in V_j$.
При построении графа конденсаций можем выделить следующие шаги:

1. Определить для каждой вершины v множество достижимых из неё вершин $R(v)$.
2. Выделить наибольшие по включению классы вершин $V_1, V_2, \ldots, V_k$, достижимых друг из друга. Эти классы будут вершинами графа G^c.
3. Построить множество дуг графа конденсации по определению.

Множество вершин, содержащих ровно одну вершину из множества новых вершин $\{V_1, V_2, \ldots, V_k\}$, имеющих нулевую полустепень захода, образует базу.

3 Расстояния в графе

3.1 Определения

Длиной $l(v_1, v_k)$ пути $P = (v_1, \{v_1, v_2\}, v_2, \{v_2, v_3\}, v_3, \ldots, v_{k-2}, \{v_{k-2}, v_{k-1}\}, v_{k-1}, \{v_{k-1}, v_k\}, v_k)$ называется суммарный вес рёбер, входящих в путь.

$$l(v_1, v_k) = \sum_i \omega_i.$$

Расстоянием $\rho(u, v)$ между двумя вершинами v и u называется длина кратчайшего пути, соединяющего эти вершины.

$$\rho(u, v) = \min_i l(u, v).$$

Эксцентриситетом $\varepsilon(v)$ вершины v называют расстояние до максимально удаленной от нее вершины. Для графа, у которого не определен вес его ребер, расстояние определяется в виде числа ребер.

$$\varepsilon(v) = \max_i \rho(u_i, v).$$

Радиус $r(G)$ графа $G = (V, E)$ – минимальный эксцентриситет его вершин.

$$r(G) = \min_i \varepsilon_i.$$

Диаметр $d(G)$ графа $G = (V, E)$ – максимальный эксцентриситет его вершин.

$$d(G) = \max_i \varepsilon_i.$$

Центром $O(G)$ связного графа $G = (V, E)$ называют множество вершин, у которых эксцентриситет равен радиусу графа.

$$O(G) = \{v | v \in V, \varepsilon(v) = r(G)\}.$$

Пример.
Дан граф $G = (V, E)$ (рисунок 7).
Пусть веса всех его рёбер равны единице.
Внесём расстояния между вершинами в таблицу.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	$\varepsilon(v_i)$
v_1	0	1	1	1	1	2	2	2
v_2	1	0	2	1	1	3	3	3
v_3	1	2	0	2	2	1	1	2
v_4	1	1	2	0	1	3	3	3
v_5	1	1	2	1	0	3	3	3
v_6	2	3	1	3	3	0	1	3
v_7	2	3	1	3	3	1	0	3

Тогда

$$d(G) = 3,$$
$$r(G) = 2,$$
$$O(G) = \{v_1, v_3\}.$$

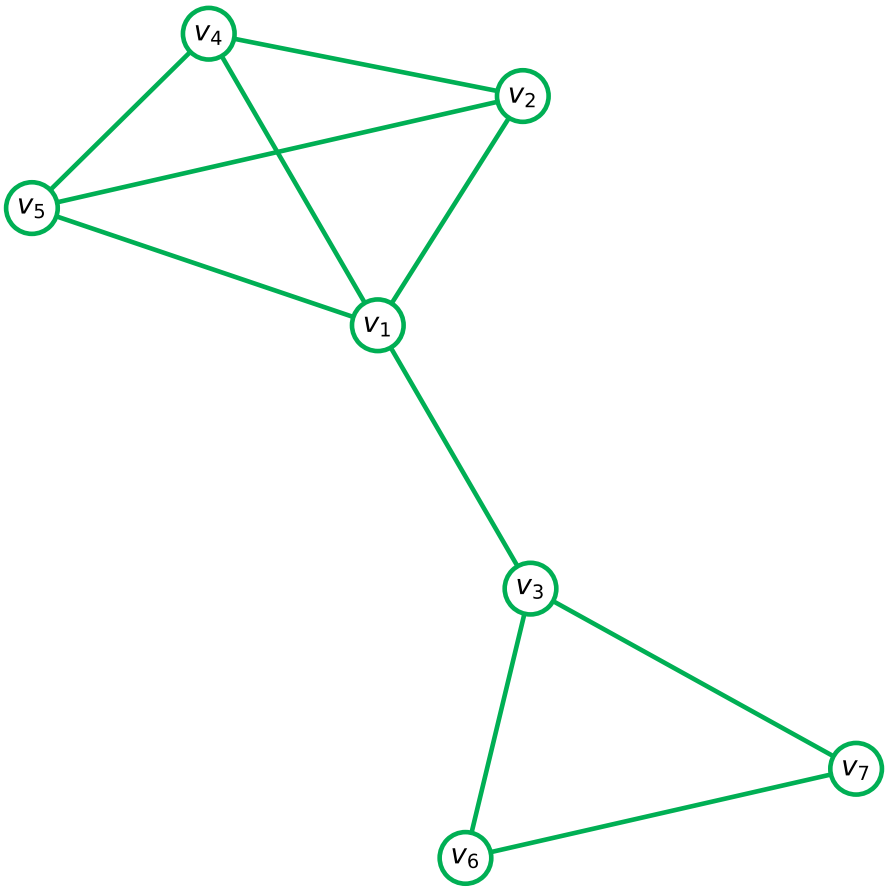

Рис. 7: Граф $G = (V, E)$

3.2 Алгоритм Дейкстры

Алгоритм позволяет построить кратчайшие пути от корневой вершины до остальных вершин графа.
Алгоритм работает на как на неориентированных, так и на ориентированных графах (в этом случае учитывается направление) в том случае, если все веса графа неотрицательны.
Описание алгоритма.

1. Корневой вершине приписывают метку $\mu_1 = 0$, а остальным вершинам метку ∞.
2. На i-м шаге выбирают вершину v_i, метка которой минимальна. Метку этой вершины называют постоянной.
3. Перебирают все смежные с v_i вершины, метка которых не является постоянной.
4. Если вершина v_j смежна с v_i и имеет метку μ_j, тогда метку вершины v_j изменяют по правилу
$$\mu_j = \min(\mu_i + \rho(v_i, v_j), \mu_j).$$
5. Алгоритм заканчивает работу, когда все вершины имеют постоянные метки.

Постоянную метку будем выделять прямоугольником, а в качестве индекса будем указывать вершину-предка. Так, запись $\boxed{1_{v_1}}$ говорит о том, что значение метки равно единице, а вершина-предок в дереве кратчайших путей – v_1.

Пример.

Дан граф $G = (V, E)$ (рисунок 8).

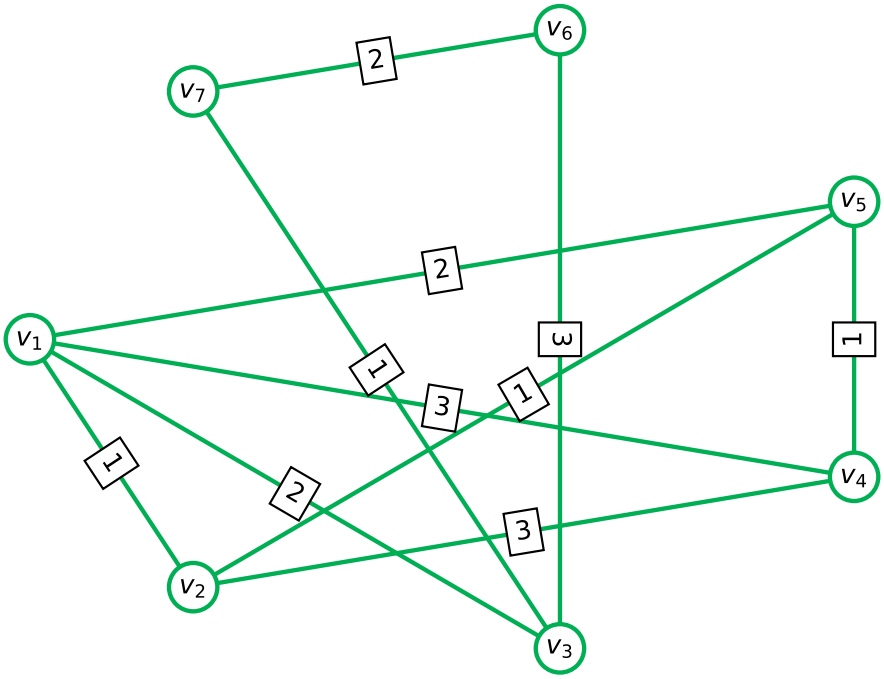

Рис. 8: Граф $G = (V, E)$

Пусть v_1 – корневая вершина, выполним первый шаг алгоритма

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
Шаг 1	$\boxed{0}$	∞	∞	∞	∞	∞	∞

Вершина v_1 смежна с v_2, v_3, v_4 и v_5.

$$\mu_2 = \min(\mu_1 + \rho(v_1, v_2), \mu_2) = \min(0 + 1, \infty) = 1,$$
$$\mu_3 = \min(\mu_1 + \rho(v_1, v_3), \mu_3) = \min(0 + 2, \infty) = 2,$$
$$\mu_4 = \min(\mu_1 + \rho(v_1, v_4), \mu_4) = \min(0 + 3, \infty) = 3,$$
$$\mu_5 = \min(\mu_1 + \rho(v_1, v_5), \mu_5) = \min(0 + 2, \infty) = 2.$$

Минимальная метка у вершины v_2, она становится постоянной.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
Шаг 1	$\boxed{0}$	∞	∞	∞	∞	∞	∞
Шаг 2		$\boxed{1_{v_1}}$	2_{v_1}	3_{v_1}	2_{v_1}	∞	∞

Вершина v_2 смежна с v_4 и v_5.

$$\mu_4 = \min(\mu_2 + \rho(v_2, v_4), \mu_4) = \min(1 + 3, 3) = 3,$$
$$\mu_5 = \min(\mu_2 + \rho(v_2, v_5), \mu_5) = \min(1 + 1, 2) = 2.$$

Минимальная метка у вершины v_3, она становится постоянной.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
Шаг 1	$\boxed{0}$	∞	∞	∞	∞	∞	∞
Шаг 2		$\boxed{1_{v_1}}$	2_{v_1}	3_{v_1}	2_{v_1}	∞	∞
Шаг 3			$\boxed{2_{v_1}}$	3_{v_1}	2_{v_1}	∞	∞

Вершина v_3 смежна с v_6 и v_7.

$$\mu_6 = \min(\mu_3 + \rho(v_3, v_6), \mu_6) = \min(2 + 3, \infty) = 5,$$
$$\mu_7 = \min(\mu_3 + \rho(v_3, v_7), \mu_7) = \min(2 + 1, \infty) = 3.$$

Минимальная метка у вершины v_5, она становится постоянной.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
Шаг 1	$\boxed{0}$	∞	∞	∞	∞	∞	∞
Шаг 2		$\boxed{1_{v_1}}$	2_{v_1}	3_{v_1}	2_{v_1}	∞	∞
Шаг 3			$\boxed{2_{v_1}}$	3_{v_1}	2_{v_1}	∞	∞
Шаг 4				3_{v_1}	$\boxed{2_{v_1}}$	5_{v_3}	3_{v_3}

Вершина v_5 смежна с v_4.

$$\mu_4 = \min(\mu_5 + \rho(v_5, v_4), \mu_4) = \min(2 + 1, 3) = 3,$$

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
Шаг 1	$\boxed{0}$	∞	∞	∞	∞	∞	∞
Шаг 2		$\boxed{1_{v_1}}$	2_{v_1}	3_{v_1}	2_{v_1}	∞	∞
Шаг 3			$\boxed{2_{v_1}}$	3_{v_1}	2_{v_1}	∞	∞
Шаг 4				3_{v_1}	$\boxed{2_{v_1}}$	5_{v_3}	3_{v_3}
Шаг 5				$\boxed{3_{v_1}}$		5_{v_3}	3_{v_3}

Вершина v_4 не смежна ни с одной вершиной с непостоянной меткой. Минимальная метка у вершины v_7, она становится постоянной.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
Шаг 1	$\boxed{0}$	∞	∞	∞	∞	∞	∞
Шаг 2		$\boxed{1_{v_1}}$	2_{v_1}	3_{v_1}	2_{v_1}	∞	∞
Шаг 3			$\boxed{2_{v_1}}$	3_{v_1}	2_{v_1}	∞	∞
Шаг 4				3_{v_1}	$\boxed{2_{v_1}}$	5_{v_3}	3_{v_3}
Шаг 5				$\boxed{3_{v_1}}$		5_{v_3}	3_{v_3}
Шаг 6						5_{v_3}	$\boxed{3_{v_3}}$

Вершина v_7 смежна с v_6.

$$\mu_6 = \min(\mu_7 + \rho(v_7, v_6), \mu_6) = \min(3 + 2, 5) = 5.$$

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
Шаг 1	$\boxed{0}$	∞	∞	∞	∞	∞	∞
Шаг 2		$\boxed{1_{v_1}}$	2_{v_1}	3_{v_1}	2_{v_1}	∞	∞
Шаг 3			$\boxed{2_{v_1}}$	3_{v_1}	2_{v_1}	∞	∞
Шаг 4				3_{v_1}	$\boxed{2_{v_1}}$	5_{v_3}	3_{v_3}
Шаг 5				$\boxed{3_{v_1}}$		5_{v_3}	3_{v_3}
Шаг 6						5_{v_3}	$\boxed{3_{v_3}}$
Шаг 7						$\boxed{5_{v_3}}$	

Таким образом, значения постоянных меток указывают на кратчайшее расстояние от корневой вершины, а двигаясь в обратном направлении по меткам от потомков к предкам, можно восстановить дерево кратчайших путей.
Для вершины v_7 постоянная метка равна 3_{v_3}, следовательно, кратчайший путь из v_1 в v_7 имеет длину 3. Так как для вершины v_7 предком является v_3, а для v_3 предком является v_1, кратчайший путь имеет вид

$$P = (v_1, \{v_1, v_3\}, v_3, \{v_3, v_7\}, v_7),$$
$$\rho(v_1, v_3) + \rho(v_3, v_7) = 2 + 1 = 3.$$