

# Dijkstra Profi

Graph Algorithm Speed Help

start from: 0



Shift + F10

V	0	1	2	3	4	5	6	7
distance(start, v)	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$

Skip back

Shift + F9

Step back

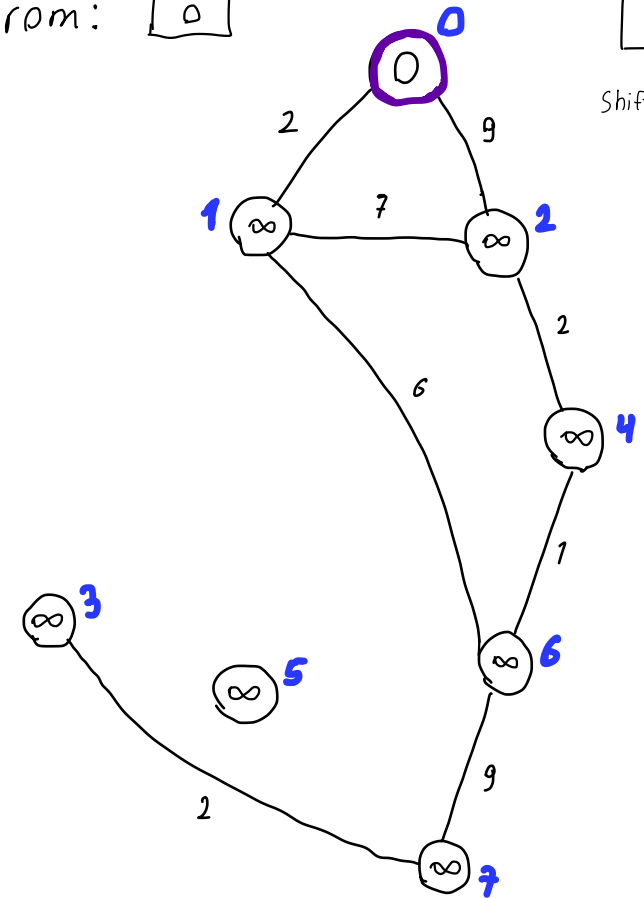
F7

Step forward

F8

Skip forward

Shift + F11



Current algorithm step

Russian

Calculations

distance(start, v) - длина кратчайшего из известных нам на данный момент путей от начальной вершины start до вершины v. По ходу работы алгоритма значения в distance будут обновляться. Изначально distance(start, start) = 0, а для всех остальных вершин длина пути равна бесконечности.

Синим цветом обозначим номера вершин. Внутри каждой вершины v будем записывать текущее значение distance(start, v).

Обработанными будем называть вершины, для которых distance больше не будет обновляться, то есть те, для которых мы уже нашли кратчайший возможный путь. В начале, конечно, все вершины необработанные.

# Dijkstra Profi

Graph Algorithm Speed Help

start from: 0



Shift + F10

V	0	1	2	3	4	5	6	7
distance(start, v)	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$

Skip back

Shift + F9

Step back

F7

Step forward

F8

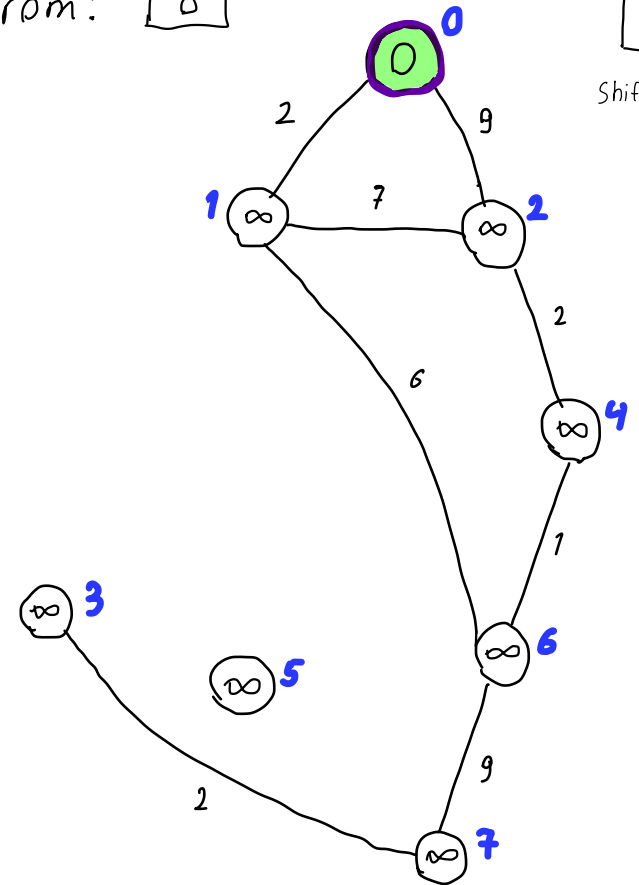
Skip forward

Shift + F11

Current algorithm step

Russian

Выбирается необработанная вершина  $v$  с минимальным расстоянием от начальной и помечается как обработанная (красится в зелёный цвет). На первом шаге - это сама начальная вершина.



Calculations

# Dijkstra Profi

Graph Algorithm Speed Help

start from: 0



Shift + F10

V	0	1	2	3	4	5	6	7
distance(start, v)	0	2	9	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$

Skip back

Shift + F9

Step back

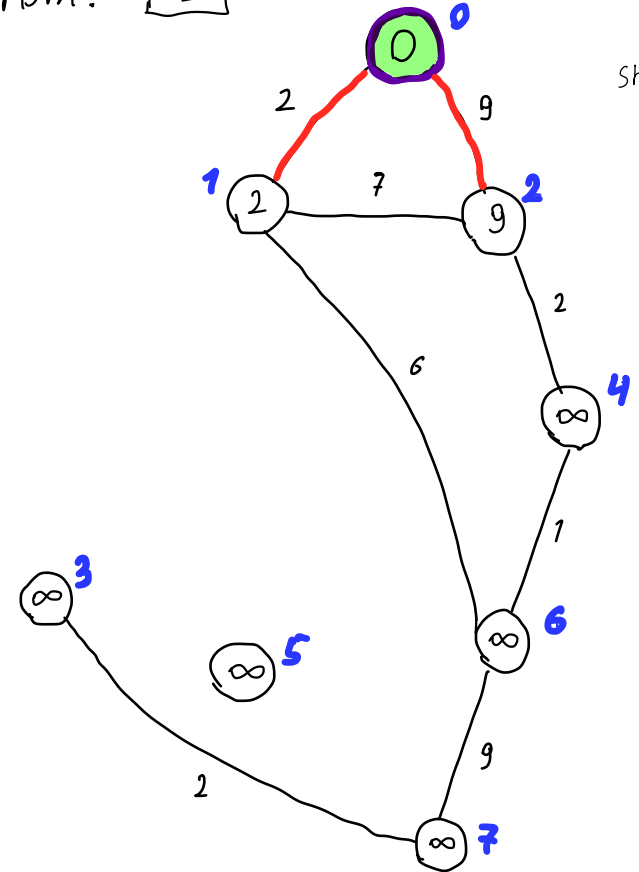
F7

Step forward

F8

Skip forward

Shift + F11



Current algorithm step

Russian

Напомним,  $v$  - вершина, которую только что поместили как обработанную. Теперь производится «релаксация»: рассматриваем все соседние с вершиной  $v$  вершины  $u$ . Вес ребра  $(v, u)$  обозначим за  $len$ . Для каждой необработанной вершины  $u$  пытаемся улучшить значение  $distance(start, u)$ . То есть, записываем в  $distance(start, u)$  минимум из текущего значения  $distance(start, u)$  и  $distance(start, v) + len$ .

Calculations

$$u = 1: \min(\infty, 0 + 2) = 2$$

$$u = 2: \min(\infty, 0 + 9) = 9$$

# Dijkstra Profi

Graph Algorithm Speed Help

start from: 0



Shift + F10

V	0	1	2	3	4	5	6	7
distance(start, v)	0	2	9	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$

Skip back

Shift + F9

Step back

F7

Step forward

F8

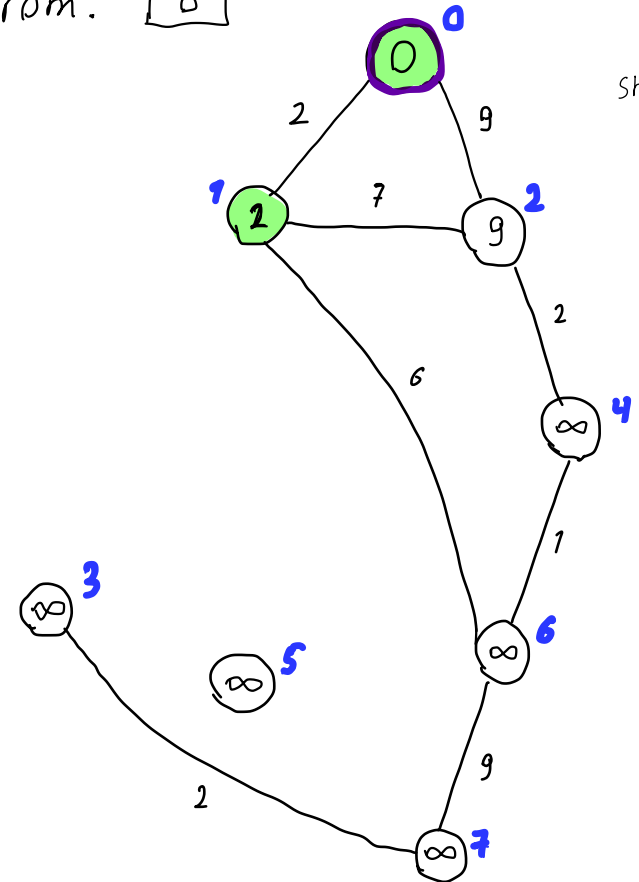
Skip forward

Shift + F11

Current algorithm step

Russian

Выбирается необработанная вершина v с минимальным расстоянием от начальной и помечается как обработанная (красится в зелёный цвет).



Calculations

# Dijkstra Profi

Graph Algorithm Speed Help

start from: 0



Shift + F10

V	0	1	2	3	4	5	6	7
distance(start, v)	0	2	9	$\infty$	$\infty$	$\infty$	8	$\infty$

Skip back

Shift + F9

Step back

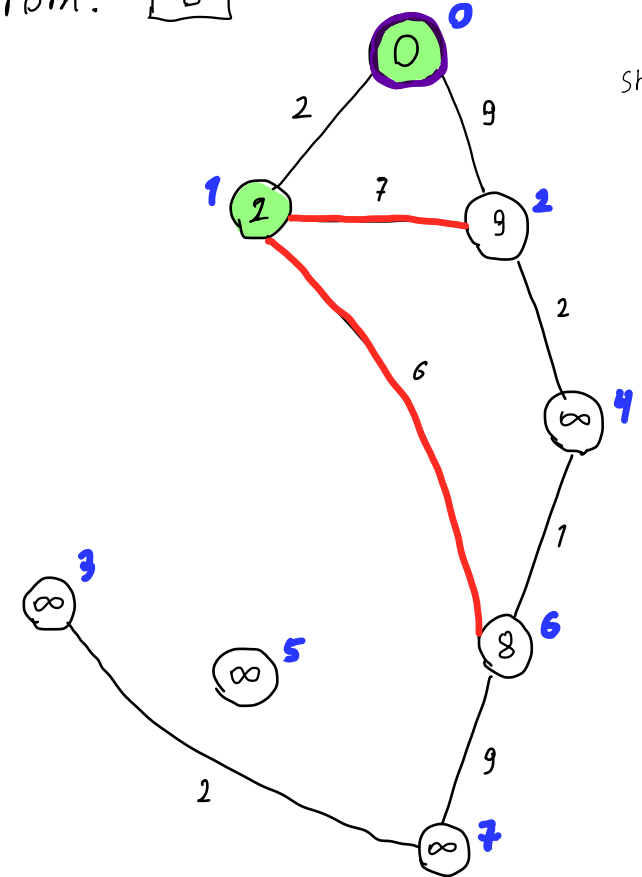
F7

Step forward

F8

Skip forward

Shift + F11



Current algorithm step

Russian

Напомним,  $v$  - вершина, которую только что поместили как обработанную. Теперь производится «релаксация»: рассматриваем все соседние с вершиной  $v$  вершины  $u$ . Вес ребра  $(v, u)$  обозначим за  $len$ . Для каждой необработанной вершины  $u$  пытаемся улучшить значение  $distance(start, u)$ . То есть, записываем в  $distance(start, u)$  минимум из текущего значения  $distance(start, u)$  и  $distance(start, v) + len$ .

Calculations

$$u = 2: \min(9, 2 + 7) = 9$$

$$u = 6: \min(\infty, 2 + 6) = 8$$

# Dijkstra Profi

Graph Algorithm Speed Help

start from: 0



Shift + F10

V	0	1	2	3	4	5	6	7
distance(start, v)	0	2	9	$\infty$	$\infty$	$\infty$	8	$\infty$

Skip back

Shift + F9

Step back

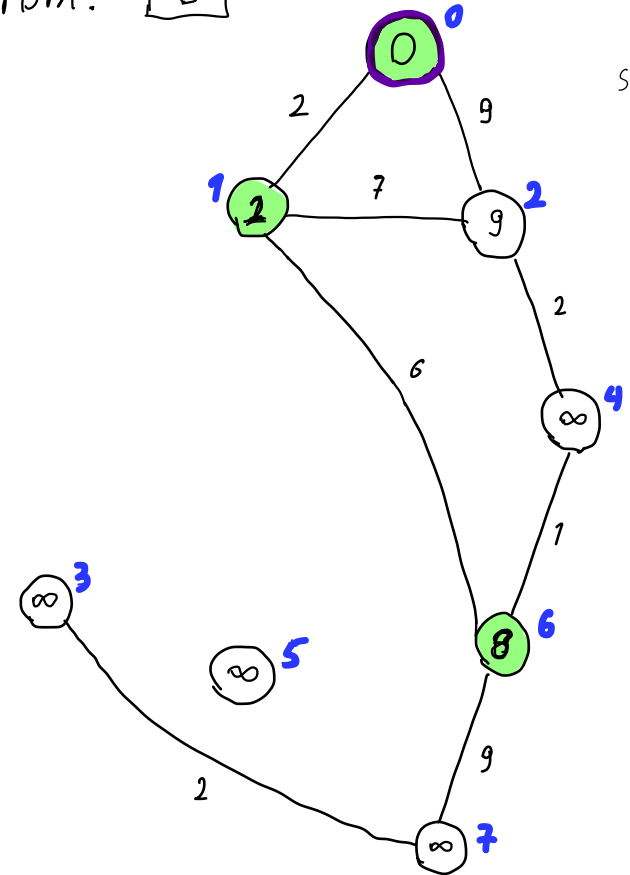
F7

Step forward

F8

Skip forward

Shift + F11



Current algorithm step

Russian

Calculations

Выбирается необработанная вершина  $v$  с минимальным расстоянием от начальной и помечается как обработанная (красится в зелёный цвет).