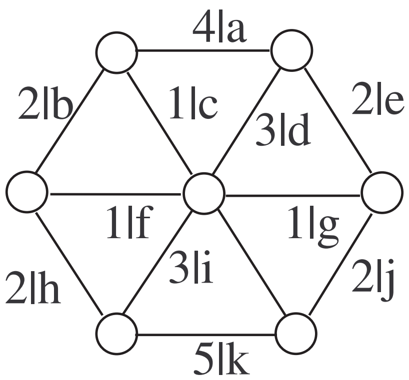
DER ALGORITHMUS VON KRUSKAL

Der Algorithmus von Kruskal

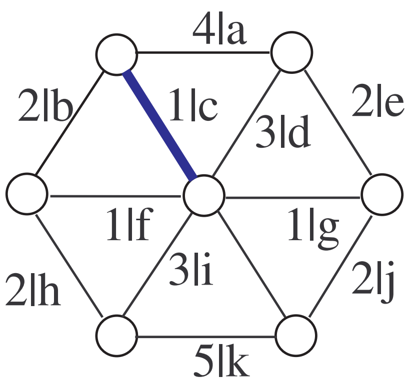


1. Kanten nach steigendem Gewicht sortieren; $E' := \emptyset$; $j := 1$;

2. `if` $(V, E' \cup \{e_j\})$ kreisfrei `then` $E' := E' \cup \{e_j\}$: `end`;

3. `If` $(|E'| = |V| - 1$ `or` $j = m)$ `then END`
   `else` $j := j + 1$; `goto` 2;
   `end`;

Der Algorithmus von Kruskal



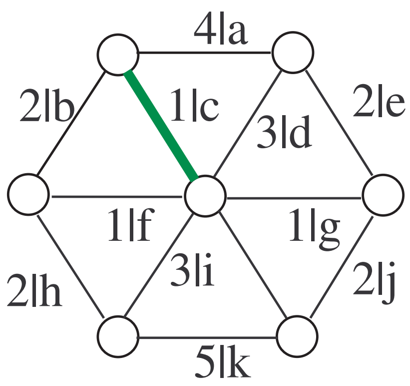
$E = \{c, f, g, b, e, h, j, d, i, a, k\}$
$E' = \emptyset$
$j = 1$

1. Kanten nach steigendem Gewicht sortieren; $E' := \emptyset$; $j := 1$;

2. `if` $(V, E' \cup \{e_j\})$ kreisfrei `then` $E' := E' \cup \{e_j\}$: `end`;

3. `If` $(|E'| = |V| - 1$ `or` $j = m)$ `then END`
   `else` $j := j + 1$; `goto` 2;
   `end`;

Der Algorithmus von Kruskal



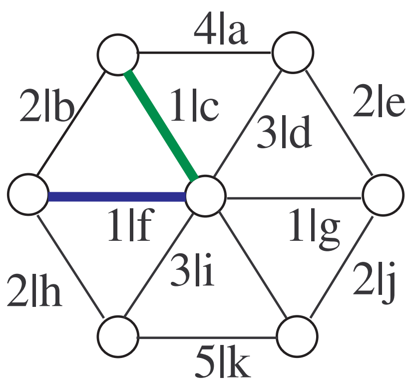
$E = \{c, f, g, b, e, h, j, d, i, a, k\}$
$E' = \{c\}$
$j = 1$

1. Kanten nach steigendem Gewicht sortieren; $E' := \emptyset$; $j := 1$;

2. `if` $(V, E' \cup \{e_j\})$ kreisfrei `then` $E' := E' \cup \{e_j\}$: `end`;

3. `If` $(|E'| = |V| - 1$ `or` $j = m)$ `then END`
   `else` $j := j + 1$; `goto` 2;
   `end`;

Der Algorithmus von Kruskal



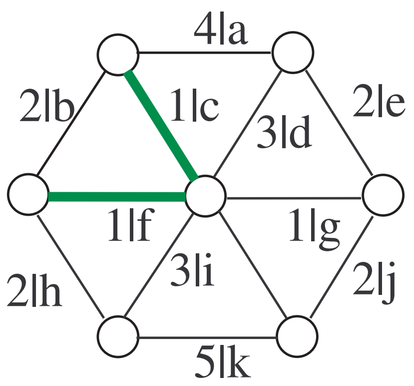
$E = \{c, f, g, b, e, h, j, d, i, a, k\}$
$E' = \{c\}$
$j = 2$

1. Kanten nach steigendem Gewicht sortieren; $E' := \emptyset$; $j := 1$;

2. `if` $(V, E' \cup \{e_j\})$ kreisfrei `then` $E' := E' \cup \{e_j\}$: `end`;

3. `If` $(|E'| = |V| - 1$ `or` $j = m)$ `then END`
    `else` $j := j + 1$; `goto` 2;
  `end`;

Der Algorithmus von Kruskal



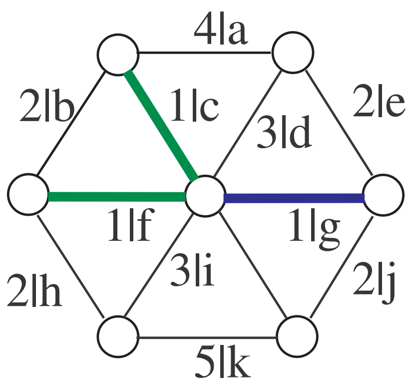
$E = \{c, f, g, b, e, h, j, d, i, a, k\}$
$E' = \{c, f\}$
$j = 2$

1. Kanten nach steigendem Gewicht sortieren; $E' := \emptyset$; $j := 1$;

2. `if` $(V, E' \cup \{e_j\})$ kreisfrei `then` $E' := E' \cup \{e_j\}$: `end`;

3. `If` $(|E'| = |V| - 1$ `or` $j = m)$ `then END`
   `else` $j := j + 1$; `goto` 2;
   `end`;

Der Algorithmus von Kruskal



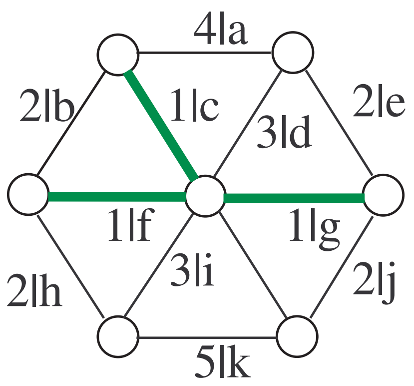
$E = \{c, f, g, b, e, h, j, d, i, a, k\}$
$E' = \{c, f\}$
$j = 3$

1. Kanten nach steigendem Gewicht sortieren; $E' := \emptyset$; $j := 1$;

2. `if` $(V, E' \cup \{e_j\})$ kreisfrei `then` $E' := E' \cup \{e_j\}$: `end`;

3. `If` $(|E'| = |V| - 1$ `or` $j = m)$ `then END`
   `else` $j := j + 1$; `goto` 2;
   `end`;

Der Algorithmus von Kruskal



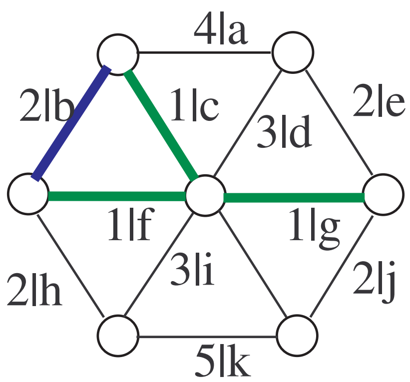
$$E = \{c, f, g, b, e, h, j, d, i, a, k\}$$
$$E' = \{c, f, g\}$$
$$j = 3$$

1. Kanten nach steigendem Gewicht sortieren; $E' := \emptyset$; $j := 1$;

2. `if` $(V, E' \cup \{e_j\})$ kreisfrei `then` $E' := E' \cup \{e_j\}$: `end`;

3. `If` $(|E'| = |V| - 1$ `or` $j = m)$ `then END`
   `else` $j := j + 1$; `goto` 2;
   `end`;

Der Algorithmus von Kruskal



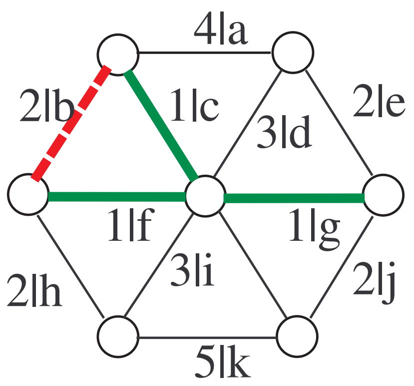
$E = \{c, f, g, b, e, h, j, d, i, a, k\}$
$E' = \{c, f, g\}$
$j = 4$

1. Kanten nach steigendem Gewicht sortieren; $E' := \emptyset$; $j := 1$;

2. `if` $(V, E' \cup \{e_j\})$ kreisfrei `then` $E' := E' \cup \{e_j\}$: `end`;

3. `If` $(|E'| = |V| - 1$ `or` $j = m)$ `then END`
   `else` $j := j + 1$; `goto` 2;
   `end`;

Der Algorithmus von Kruskal



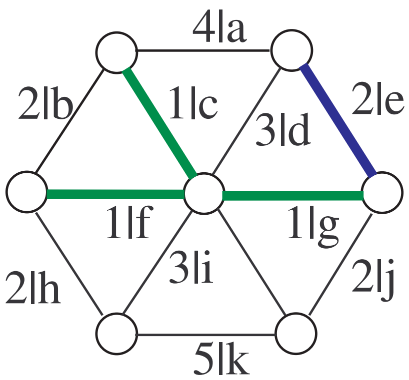
$E = \{c, f, g, b, e, h, j, d, i, a, k\}$
$E' = \{c, f, g\}$
$j = 4$

1. Kanten nach steigendem Gewicht sortieren; $E' := \emptyset$; $j := 1$;

2. `if` $(V, E' \cup \{e_j\})$ kreisfrei `then` $E' := E' \cup \{e_j\}$: `end`;

3. `If` $(|E'| = |V| - 1$ `or` $j = m)$ `then END`
   `else` $j := j + 1$; `goto` 2;
   `end`;

Der Algorithmus von Kruskal



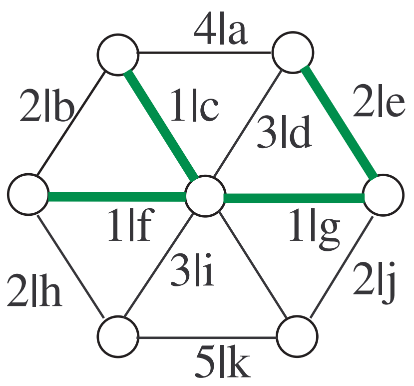
$E = \{c, f, g, b, e, h, j, d, i, a, k\}$
$E' = \{c, f, g\}$
$j = 5$

1. Kanten nach steigendem Gewicht sortieren; $E' := \emptyset$; $j := 1$;

2. `if` $(V, E' \cup \{e_j\})$ kreisfrei `then` $E' := E' \cup \{e_j\}$: `end`;

3. `If` $(|E'| = |V| - 1$ `or` $j = m)$ `then END`
   `else` $j := j + 1$; `goto` 2;
   `end`;

Der Algorithmus von Kruskal



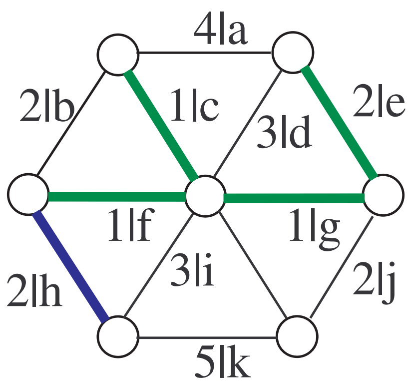
$E = \{c, f, g, b, e, h, j, d, i, a, k\}$
$E' = \{c, f, g, e\}$
$j = 5$

1. Kanten nach steigendem Gewicht sortieren; $E' := \emptyset$; $j := 1$;

2. `if` $(V, E' \cup \{e_j\})$ kreisfrei `then` $E' := E' \cup \{e_j\}$: `end`;

3. `If` $(|E'| = |V| - 1$ `or` $j = m)$ `then END`
   `else` $j := j + 1$; `goto` 2;
   `end`;

Der Algorithmus von Kruskal



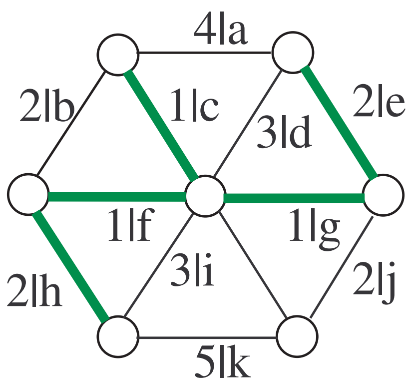
$E = \{c, f, g, b, e, h, j, d, i, a, k\}$
$E' = \{c, f, g, e\}$
$j = 6$

1. Kanten nach steigendem Gewicht sortieren; $E' := \emptyset$; $j := 1$;

2. `if` $(V, E' \cup \{e_j\})$ kreisfrei `then` $E' := E' \cup \{e_j\}$: `end`;

3. `If` $(|E'| = |V| - 1$ `or` $j = m)$ `then END`
   `else` $j := j + 1$; `goto` 2;
   `end`;

Der Algorithmus von Kruskal



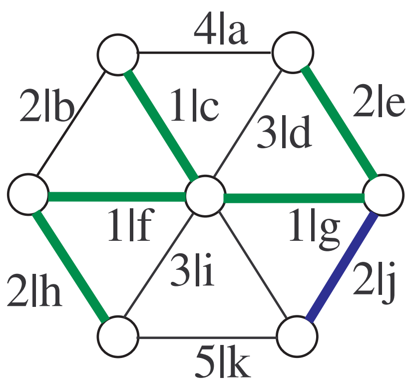
$E = \{c, f, g, b, e, h, j, d, i, a, k\}$
$E' = \{c, f, g, e, h\}$
$j = 6$

1. Kanten nach steigendem Gewicht sortieren; $E' := \emptyset$; $j := 1$;

2. `if` $(V, E' \cup \{e_j\})$ kreisfrei `then` $E' := E' \cup \{e_j\}$: `end`;

3. `If` $(|E'| = |V| - 1$ `or` $j = m)$ `then END`
   `else` $j := j + 1$; `goto` 2;
   `end`;

Der Algorithmus von Kruskal



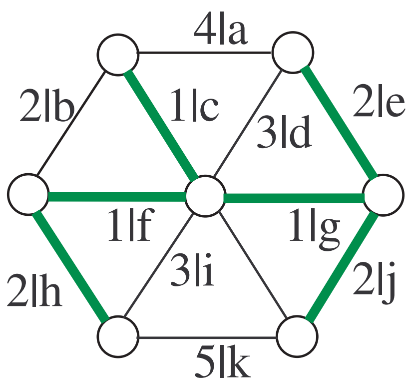
$E = \{c, f, g, b, e, h, j, d, i, a, k\}$
$E' = \{c, f, g, e, h\}$
$j = 7$

1. Kanten nach steigendem Gewicht sortieren; $E' := \emptyset$; $j := 1$;

2. **`if`** $(V, E' \cup \{e_j\})$ kreisfrei **`then`** $E' := E' \cup \{e_j\}$: **`end`**;

3. **`If`** $(|E'| = |V| - 1$ **`or`** $j = m)$ **`then END`**
   `else` $j := j + 1$; **`goto`** 2;
   `end`;

Der Algorithmus von Kruskal



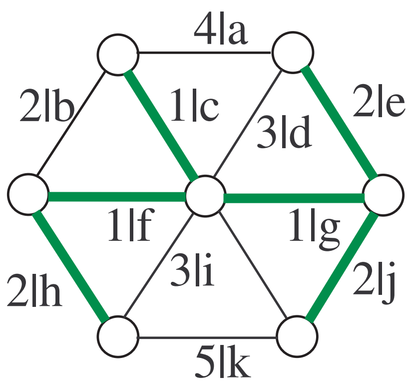
$E = \{c, f, g, b, e, h, j, d, i, a, k\}$
$E' = \{c, f, g, e, h, j\}$
$j = 7$

1. Kanten nach steigendem Gewicht sortieren; $E' := \emptyset$; $j := 1$;

2. **if** $(V, E' \cup \{e_j\})$ kreisfrei **then** $E' := E' \cup \{e_j\}$: **end**;

3. **If** $(|E'| = |V| - 1$ **or** $j = m)$ **then END**
   else $j := j + 1$; **goto** 2;
   end;

Der Algorithmus von Kruskal



$E = \{c, f, g, b, e, h, j, d, i, a, k\}$
$E' = \{c, f, g, e, h, j\}$
$j = 7$
$\boxed{|E'| = 6 \rightsquigarrow ENDE}$

1. Kanten nach steigendem Gewicht sortieren; $E' := \emptyset$; $j := 1$;

2. `if` $(V, E' \cup \{e_j\})$ kreisfrei `then` $E' := E' \cup \{e_j\}$: `end`;

3. `If` $(|E'| = |V| - 1$ `or` $j = m)$ `then END`
   `else` $j := j + 1$; `goto` 2;
   `end`;

DER ALGORITHMUS VON DIJKSTRA

Der Algorithmus von Dijkstra

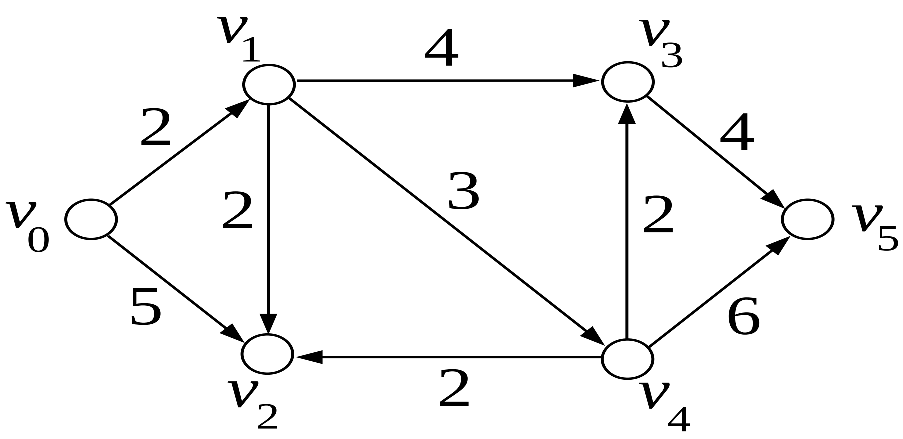
1. $l(v_0) := 0$; `for` $v \in V \setminus \{v_0\}$ `do` $l(v) := \infty$ `end`; $U := \{v_0\}$; $u := v_0$;

2. `for` $v \in V \setminus U$ `do`:
   `if` $(u, v) \in E$ `and` $l(v) > l(u) + w(u, v)$ `then`
   $p(v) := u$;
   $l(v) := l(u) + w(u, v)$;
   `end if`;

3. $m := \min\limits_{v \in V \setminus U} l(v)$, wähle Knoten $z \in V \setminus U$ mit $l(z) = m$;
   $U := U \cup \{z\}$;
   $u := z$;

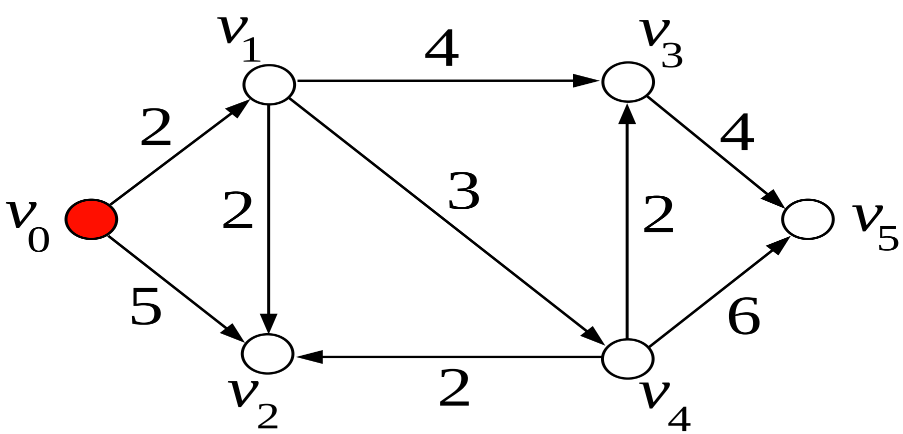
4. `if` $U = V$ `or` $\forall v \in V \setminus U \; : \; l(v) = \infty$ `then END`
   `else goto 2`;

Der Algorithmus von Dijkstra



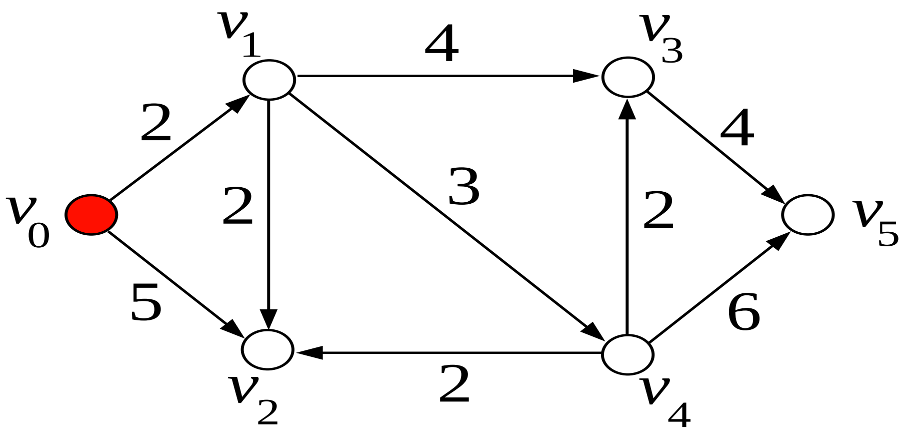
	v_0	v_1	v_2	v_3	v_4	v_5	Auswahl	Vorgänger
0	0	∞	∞	∞	∞	∞		

Der Algorithmus von Dijkstra



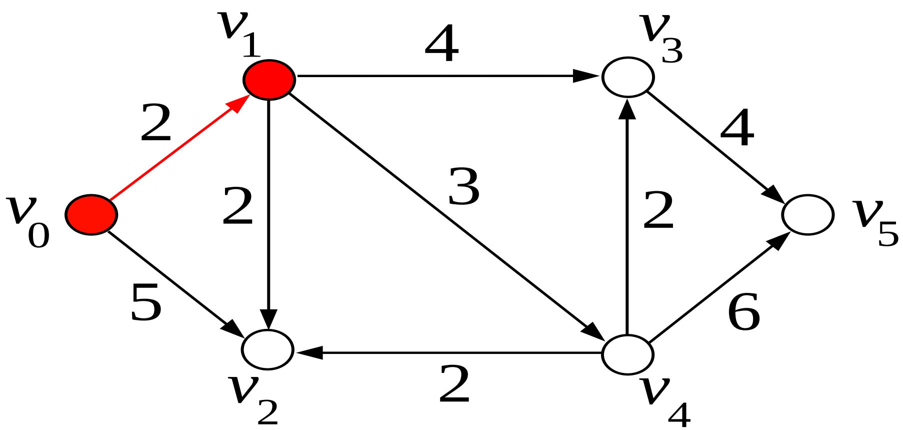
	v_0	v_1	v_2	v_3	v_4	v_5	Auswahl	Vorgänger
0	$\boxed{0}$	∞	∞	∞	∞	∞	v_0	

Der Algorithmus von Dijkstra



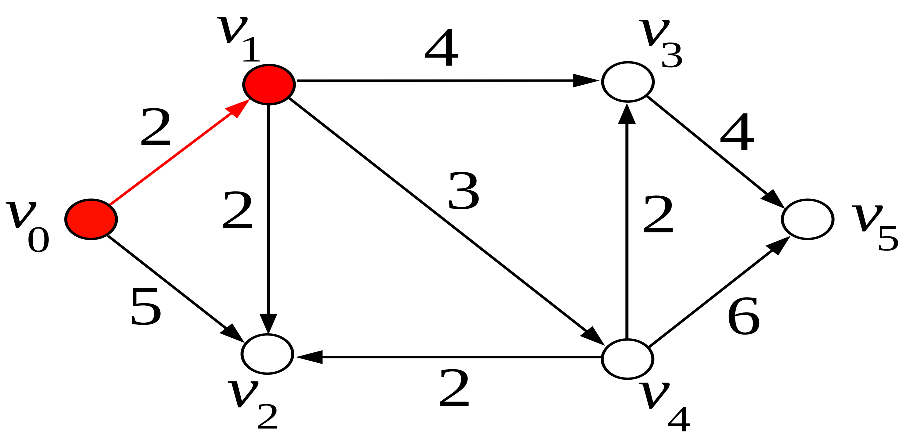
	v_0	v_1	v_2	v_3	v_4	v_5	Auswahl	Vorgänger
0	$\boxed{0}$	∞	∞	∞	∞	∞	v_0	
1		$2/v_0$	$5/v_0$	∞	∞	∞		

Der Algorithmus von Dijkstra



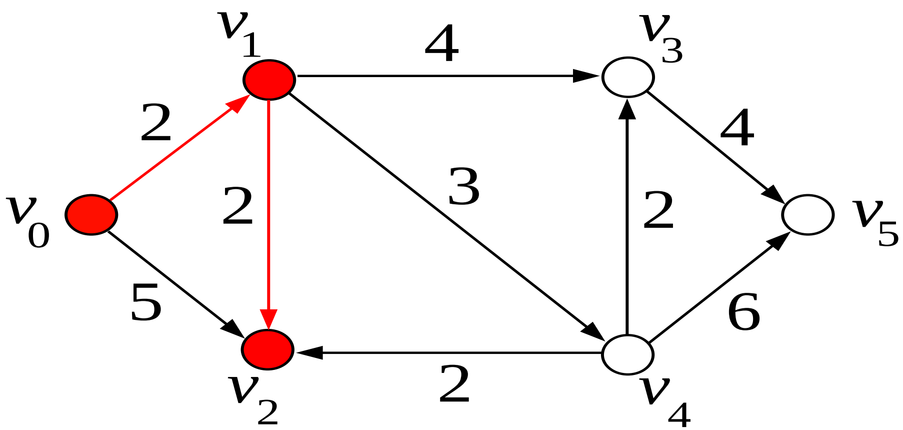
	v_0	v_1	v_2	v_3	v_4	v_5	Auswahl	Vorgänger
0	$\boxed{0}$	∞	∞	∞	∞	∞	v_0	
1		$\boxed{2/v_0}$	$5/v_0$	∞	∞	∞	v_1	v_0

Der Algorithmus von Dijkstra



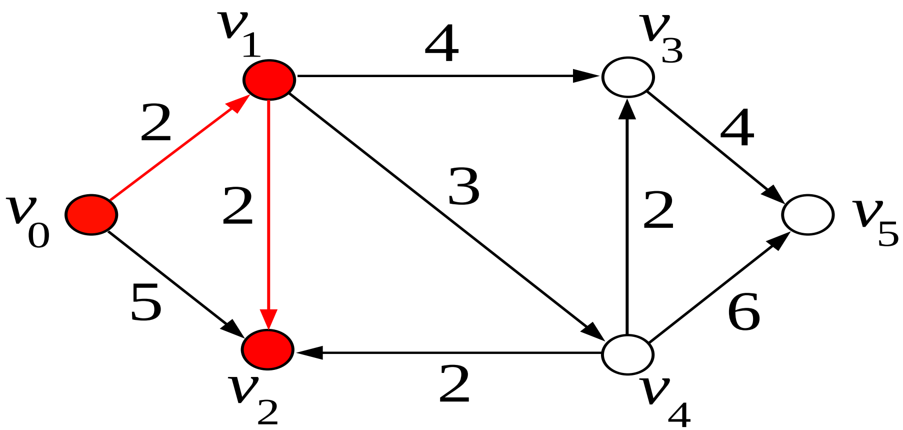
	v_0	v_1	v_2	v_3	v_4	v_5	Auswahl	Vorgänger
0	$\boxed{0}$	∞	∞	∞	∞	∞	v_0	
1		$\boxed{2/v_0}$	$5/v_0$	∞	∞	∞	v_1	v_0
2			$4/v_1$	$6/v_1$	$5/v_1$	∞		

Der Algorithmus von Dijkstra



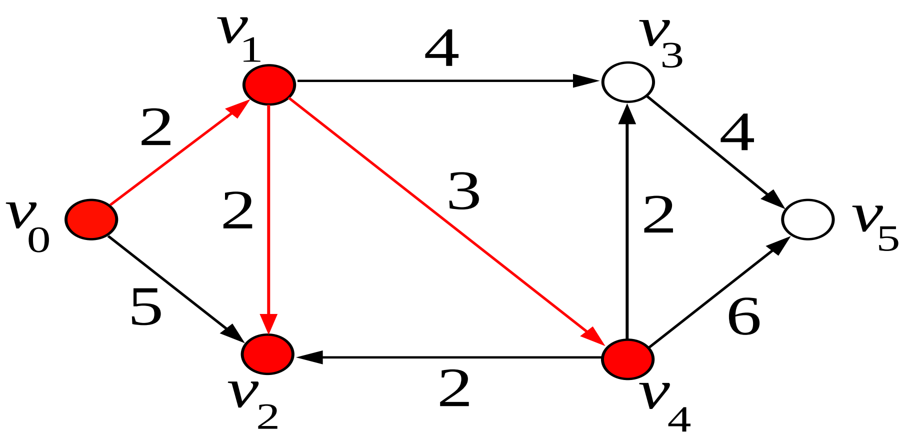
	v_0	v_1	v_2	v_3	v_4	v_5	Auswahl	Vorgänger
0	$\boxed{0}$	∞	∞	∞	∞	∞	v_0	
1		$\boxed{2/v_0}$	$5/v_0$	∞	∞	∞	v_1	v_0
2			$\boxed{4/v_1}$	$6/v_1$	$5/v_1$	∞	v_2	v_1

Der Algorithmus von Dijkstra



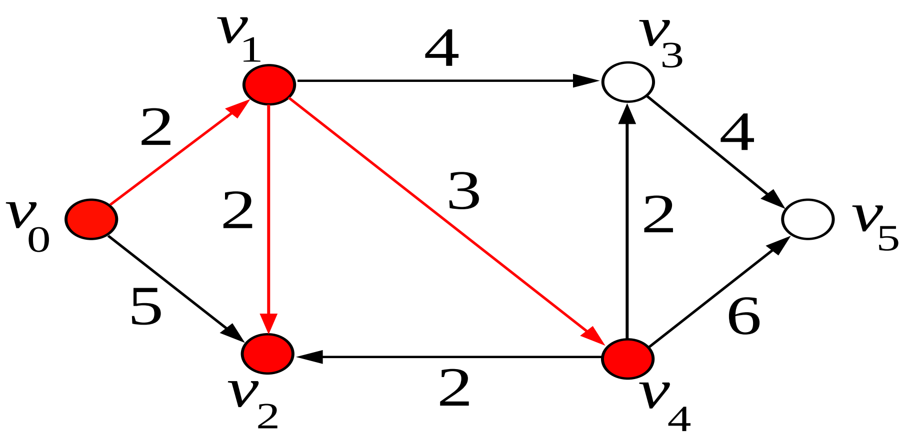
	v_0	v_1	v_2	v_3	v_4	v_5	Auswahl	Vorgänger
0	$\boxed{0}$	∞	∞	∞	∞	∞	v_0	
1		$\boxed{2/v_0}$	$5/v_0$	∞	∞	∞	v_1	v_0
2			$\boxed{4/v_1}$	$6/v_1$	$5/v_1$	∞	v_2	v_1
3				$6/v_1$	$5/v_1$	∞		

Der Algorithmus von Dijkstra



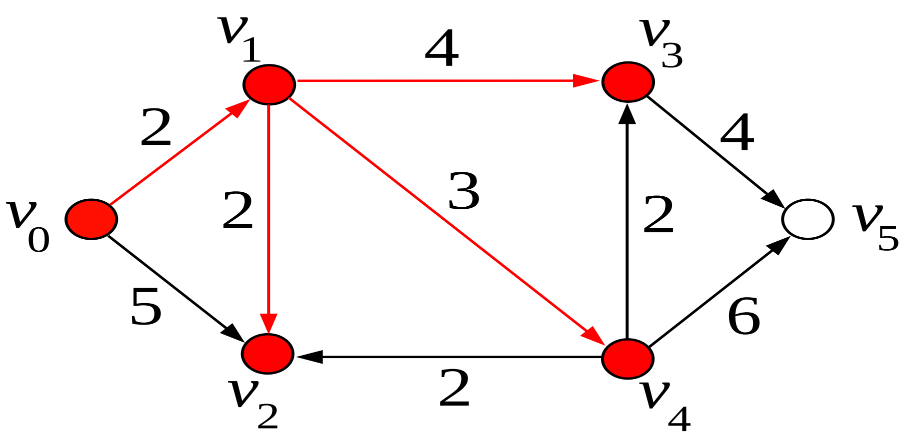
	v_0	v_1	v_2	v_3	v_4	v_5	Auswahl	Vorgänger
0	$\boxed{0}$	∞	∞	∞	∞	∞	v_0	
1		$\boxed{2/v_0}$	$5/v_0$	∞	∞	∞	v_1	v_0
2			$\boxed{4/v_1}$	$6/v_1$	$5/v_1$	∞	v_2	v_1
3				$6/v_1$	$\boxed{5/v_1}$	∞	v_4	v_1

Der Algorithmus von Dijkstra



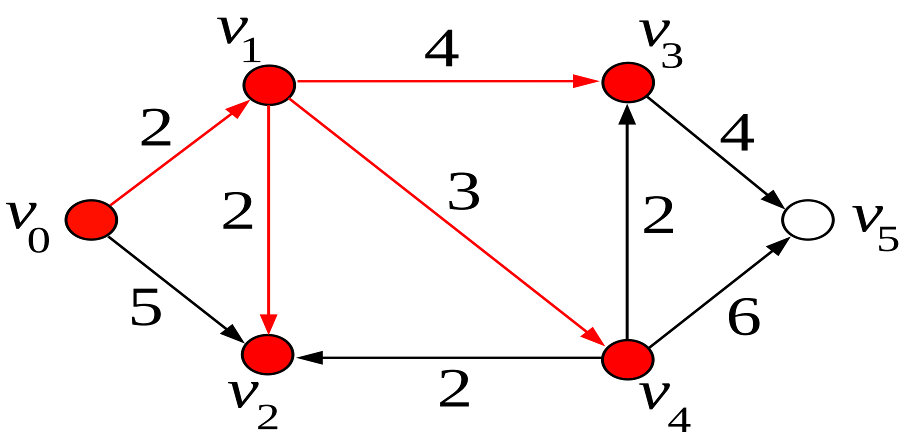
	v_0	v_1	v_2	v_3	v_4	v_5	Auswahl	Vorgänger
0	$\boxed{0}$	∞	∞	∞	∞	∞	v_0	
1		$\boxed{2/v_0}$	$5/v_0$	∞	∞	∞	v_1	v_0
2			$\boxed{4/v_1}$	$6/v_1$	$5/v_1$	∞	v_2	v_1
3				$6/v_1$	$\boxed{5/v_1}$	∞	v_4	v_1
4				$6/v_1$		$11/v_4$		

Der Algorithmus von Dijkstra



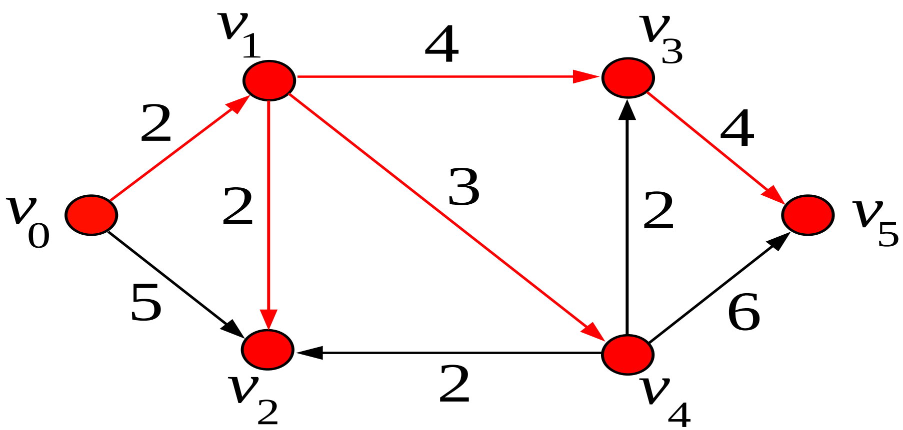
	v_0	v_1	v_2	v_3	v_4	v_5	Auswahl	Vorgänger
0	$\boxed{0}$	∞	∞	∞	∞	∞	v_0	
1		$\boxed{2/v_0}$	$5/v_0$	∞	∞	∞	v_1	v_0
2			$\boxed{4/v_1}$	$6/v_1$	$5/v_1$	∞	v_2	v_1
3				$6/v_1$	$\boxed{5/v_1}$	∞	v_4	v_1
4				$\boxed{6/v_1}$		$11/v_4$	v_3	v_1

Der Algorithmus von Dijkstra



	v_0	v_1	v_2	v_3	v_4	v_5	Auswahl	Vorgänger
0	$\boxed{0}$	∞	∞	∞	∞	∞	v_0	
1		$\boxed{2/v_0}$	$5/v_0$	∞	∞	∞	v_1	v_0
2			$\boxed{4/v_1}$	$6/v_1$	$5/v_1$	∞	v_2	v_1
3				$6/v_1$	$\boxed{5/v_1}$	∞	v_4	v_1
4				$\boxed{6/v_1}$		$11/v_4$	v_3	v_1
5						$10/v_3$		

Der Algorithmus von Dijkstra



	v_0	v_1	v_2	v_3	v_4	v_5	Auswahl	Vorgänger
0	$\boxed{0}$	∞	∞	∞	∞	∞	v_0	
1		$\boxed{2/v_0}$	$5/v_0$	∞	∞	∞	v_1	v_0
2			$\boxed{4/v_1}$	$6/v_1$	$5/v_1$	∞	v_2	v_1
3				$6/v_1$	$\boxed{5/v_1}$	∞	v_4	v_1
4				$\boxed{6/v_1}$		$11/v_4$	v_3	v_1
5						$\boxed{10/v_3}$	v_5	v_3