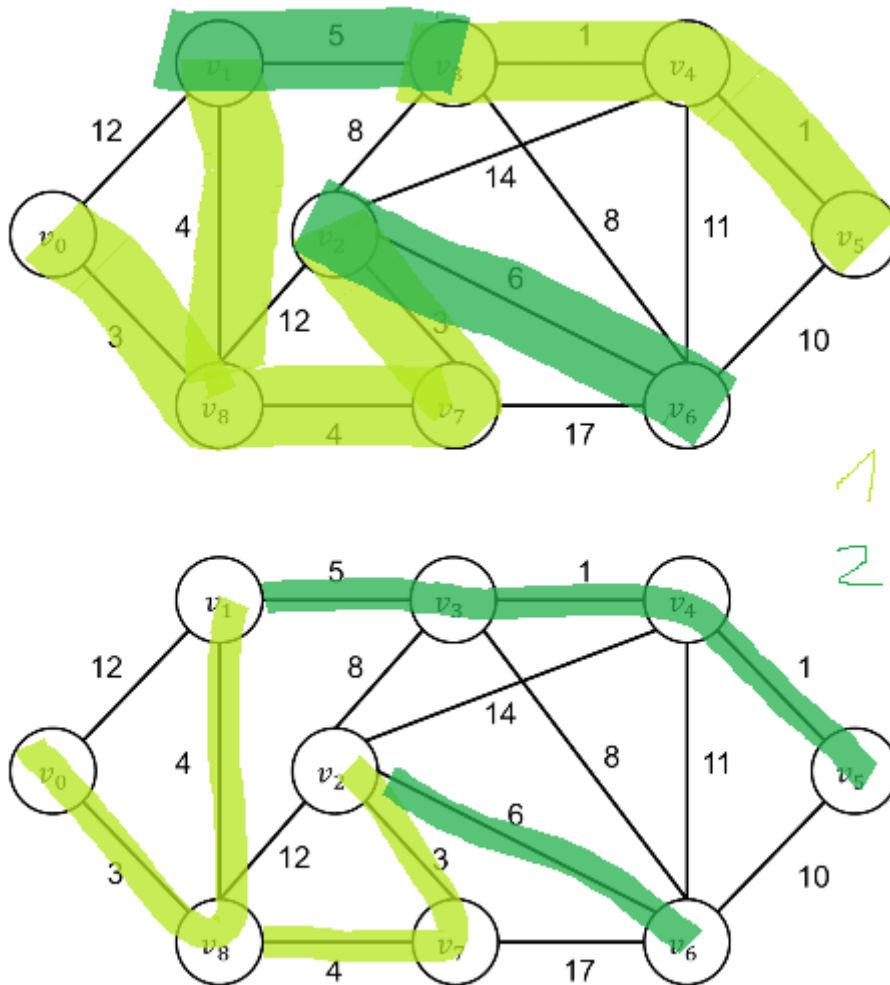


Algorithmen und Datenstrukturen Blatt 12 Aufgabe 1



Die obere Grafik zeigt das Vorgehen mit Kruskals Algorithmus und die untere Grafik Prim.

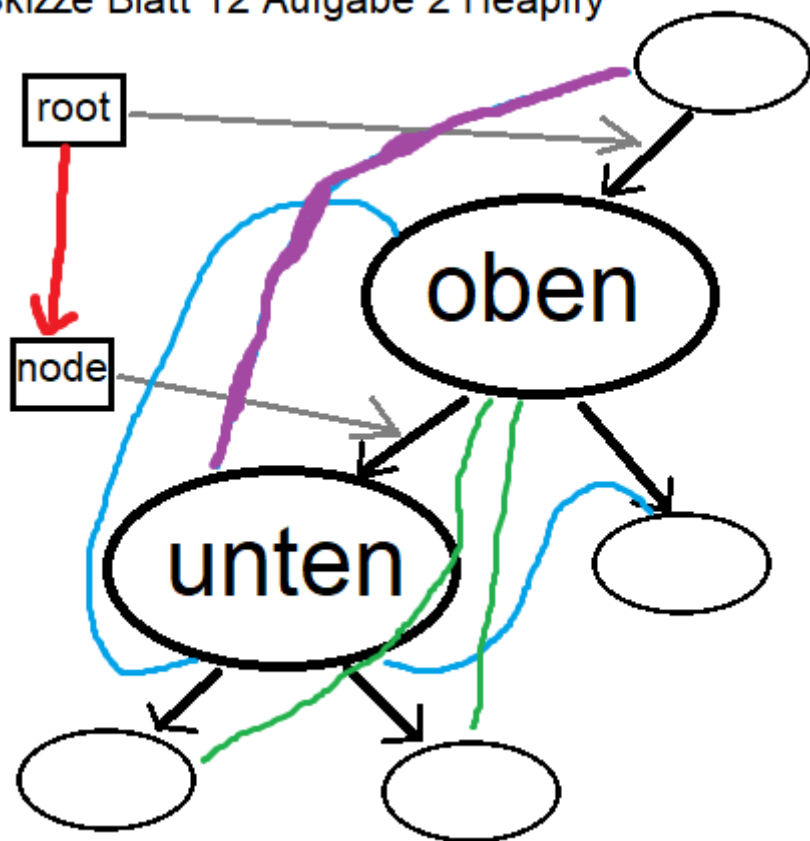
Die Einzelschritte zu Kruskal:

von	nach	Gewicht	Gesamtgewicht	Inselgroesse	Farbe in Grafik
v_3	v_4	+1	= 1	neue Insel	hellgrün, oben rechts
v_4	v_5	+1	= 2	Insel-Extension: 3	hellgrün, oben rechts
v_0	v_8	+3	= 5	neue Insel	hellgrün, unten links
v_2	v_7	+3	= 8	neue Insel	hellgrün, unten links
v_1	v_8	+4	= 12	Insel-Extension: 3	hellgrün, unten links
v_7	v_8	+4	= 16	Insel-Join: $3+2 = 5$	hellgrün, unten links
v_1	v_3	+5	= 21	Insel-Join: $5+3 = 8$	dunkelgrün, oben
v_2	v_6	+6	= 27	Insel-Extension: 9	dunkelgrün, unten

Die Einzelschritte zu Prim:

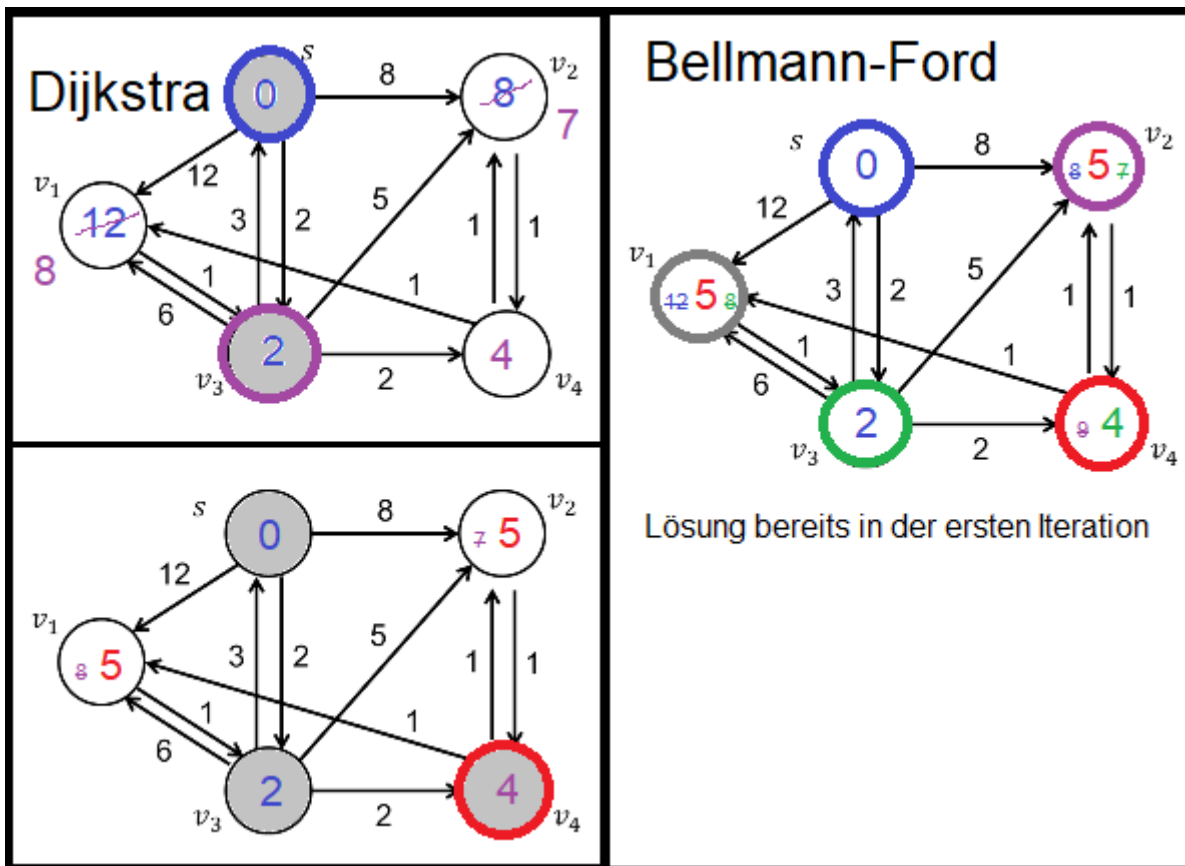
von	nach	Gewicht	Gesamtgewicht	Heap-Root	Heap-Left	Heap-Right	Farbe
v_0	v_8	+3	= 3	$v_1 : 4$	$v_2 : 12$	$v_7 : 4$	hellgrün
v_8	v_1	+4	= 7	$v_7 : 4$	$v_2 : 12$	$v_3 : 5$	hellgrün
v_8	v_7	+4	= 11	$v_2 : 3$	$v_3 : 5$	$v_6 : 17$	hellgrün
v_7	v_2	+3	= 14	$v_3 : 5$	$v_4 : 14$	$v_6 : 6$	hellgrün
v_1	v_3	+5	= 19	$v_4 : 1$	$v_6 : 6$	–	dunkelgrün
v_3	v_4	+1	= 20	$v_5 : 1$	$v_6 : 6$	–	dunkelgrün
v_4	v_5	+1	= 21	$v_6 : 6$	–	–	dunkelgrün
v_2	v_6	+6	= 27	–	–	–	dunkelgrün

Skizze Blatt 12 Aufgabe 2 Heapify



Die Grafik zeigt die Pointer-Umbiegungen beim Heapify.
Details sind als Dokumentation im Quellcode angegeben.

Algorithmen und Datenstrukturen Blatt 12 Aufgabe 3



Dijkstra-Algorithmus Step-by-Step:

s	v_1	v_2	v_3	v_4	Heap-Root	Heap-Left	Heap-Right	Farbe
0	12	8	2	x	$v_3 : 2$	$v_1 : 12$	$v_2 : 8$	blau
.	8	7		4	$v_4 : 4$	$v_1 : 8$	$v_2 : 7$	lila
	5	5			$v_1 : 5$	$v_2 : 5$	–	rot
					$v_2 : 5$	–	–	

Algorithmen und Datenstrukturen Blatt 12 Aufgabe 4a

Warum kann Dijkstra nicht mit negativen Zahlen umgehen?

Dijkstra markiert Knoten als erledigt, was bei negativen Gewichten falsch sein kann.
Lösung ist einfach: Die Erledigt-Markierungen einfach erst gar nicht implementieren.

.....

Algorithmen und Datenstrukturen Blatt 12 Aufgabe 4b

Warum kann man nicht das Minimum abziehen und so negative Zahlen eliminieren?

Graph-Skizze:

$[S] - > [A] - > [B] - > [C] < -[S]$

Adjazenz-Kosten-Matrix (von Zeile nach Spalte):

	S	A	B	C	\sum	Ergebnis
S	x	+6	-	+4	+0	
A	-	x	-3	-	+6	
B	-	-	x	-2	+3	
C	-	-	-	x	+1	langer Weg ist besser (als 4)

Subtrahiert man nun überall das Minimum, also - -3, also +3, erhält man:

	S	A	B	C	\sum	Ergebnis
S	x	+9	-	+7	+0	
A	-	x	+0	-	+9	
B	-	-	x	+1	+9	
C	-	-	-	x	+7	kurzer Weg ist besser (als 10)

Also: Modifizierter Algorithmus liefert anderes Ergebnis

Ergebnis: Algorithmus ist falsch.