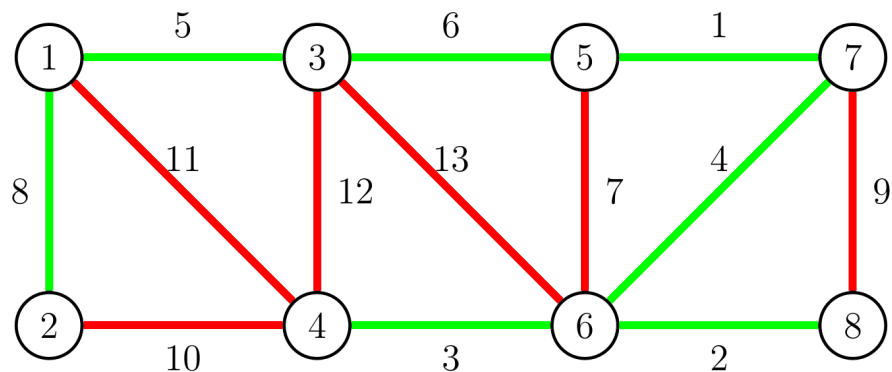


Algorithmen und Datenstrukturen 1

Übungsblatt 3

Aufgabe 18.

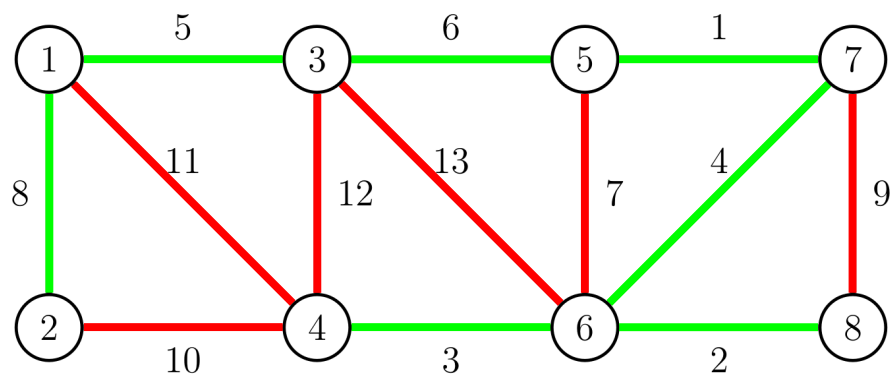
Lösung. Den Graphen aus Beispiel 19 nehmend:



Der kürzeste Weg im ursprünglichen Graphen zwischen Knoten 2 und 4 wäre mit 10 gewichtet. Im minimalen Spannbaum müsste ein Umweg (2-1-3-5-7-6-4) gemacht werden, welcher mit 26 gewichtet ist.

Aufgabe 19.

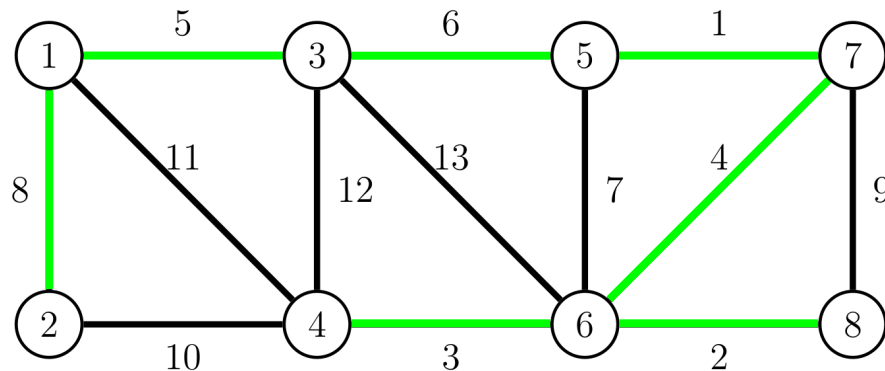
Lösung.



Die Kanten werden nach aufsteigender Kantengewichtung ausgewählt.

Aufgabe 20.

Lösung.



Ausgewählt	Priority Queue	Knotenmenge S	Gewicht
2	1, 4, 3, 5, 6, 7, 8	2	0
1	3, 4, 5, 6, 7, 8	2, 1	8
3	5, 4, 6, 7, 8	2, 1, 3	13
5	7, 6, 4, 8	2, 1, 3, 5	19
7	6, 8, 4	2, 1, 3, 5, 7	20
6	8, 4	2, 1, 3, 5, 7, 6	24
8	4	2, 1, 3, 5, 7, 6, 8	26
4		2, 1, 3, 5, 7, 6, 8, 4	29

Die Kanten wurden in der Reihenfolge $8 \rightarrow 5 \rightarrow 6 \rightarrow 1 \rightarrow 4 \rightarrow 2 \rightarrow 3$ ausgewählt.

Aufgabe 21.

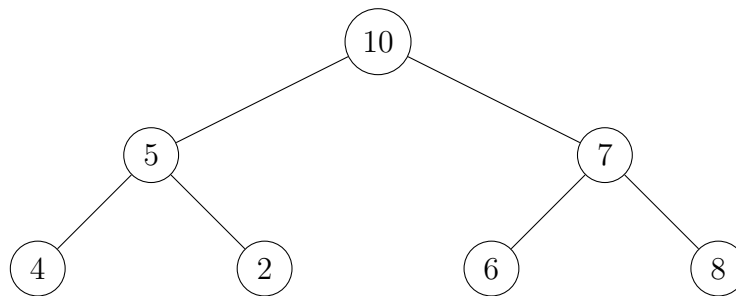
Lösung.

- Für das Finden eines maximalen Elements ist der Max-Heap besser geeignet, da sich das maximale Element an der ersten Stelle befindet. In einem sortierten Array (welcher sich als Min-Heap qualifiziert) befindet sich das maximale Element am Ende.
- Das Löschen eines gegebenen Elements funktioniert im Max-Heap besser, weil in einem sortierten Array alle Elemente um eins verschoben werden müssen.
- Das Sortieren eines Arrays gelingt nicht besser als $O(n \log n)$. Auch das Aufbauen eines Max-Heaps ist nicht schneller als $O(n \log n)$.
- Das Finden eines minimalen Elements geht schneller im sortierten Array, weil da das kleinste Element am Anfang steht. In einem Max-Heap befindet sich das kleinste Element in einem der letzten Kindknoten, wodurch zumindest diese durchsucht werden müssen.

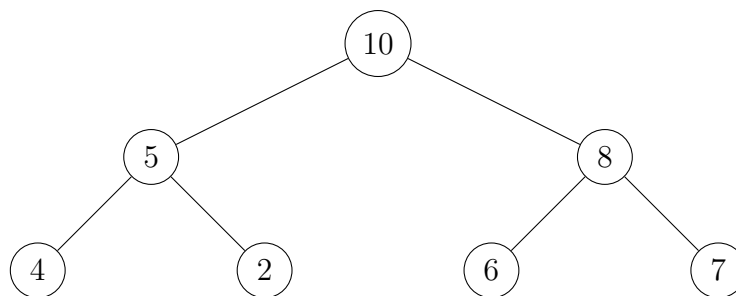
Aufgabe 22.

Lösung.

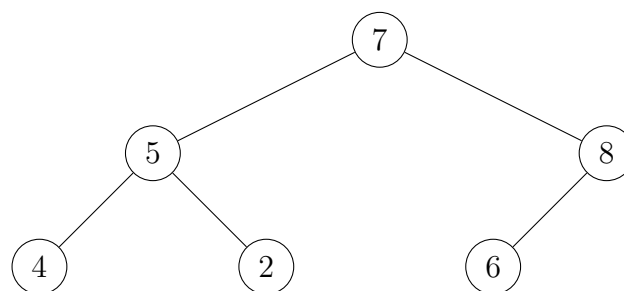
- a) Der neue Knoten wird an Position $n + 1$ eingefügt:



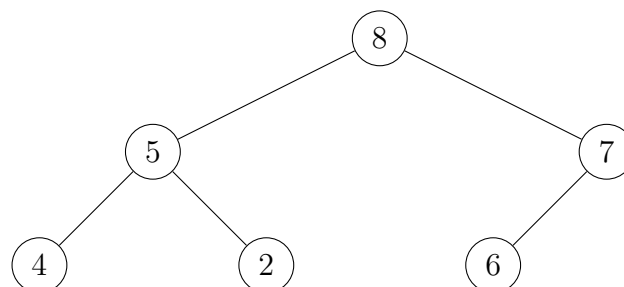
Dann wird der neue Knoten (8) mit seinem Parent-Knoten (7) vertauscht, da er größer als dieser ist:



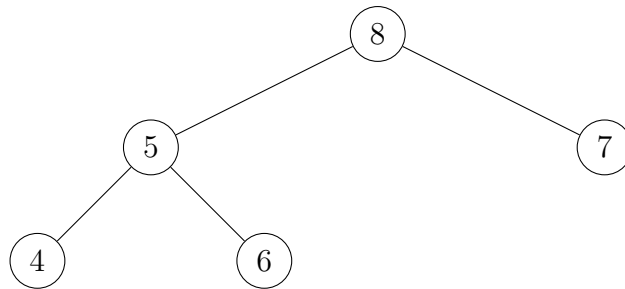
- b) Der größte Knoten (10) wird gelöscht und der Knoten an der Position n (7) wird stattdessen eingefügt.



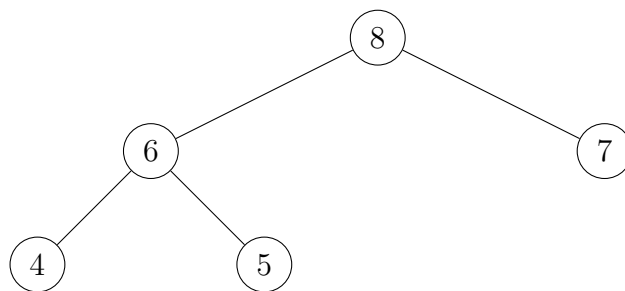
Danach muss der Knoten 7 mit seinem rechten Kindknoten 8 vertauscht werden:



Um den Knoten 2 zu löschen muss an dessen Stelle der n -te Knoten (6) geschoben werden:



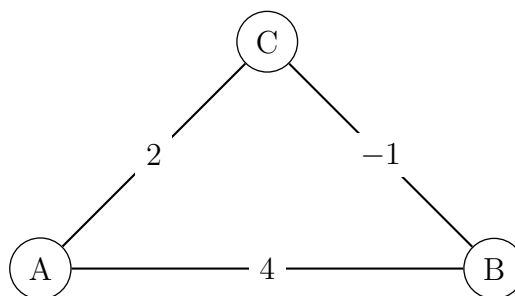
Jetzt den Knoten 6 mit seinem Parent-Knoten vertauschen:



Aufgabe 23.

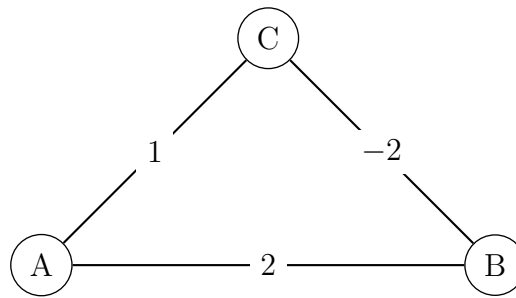
Lösung.

a)



Ausgehend vom Startknoten A findet der Dijkstra-Algorithmus den richtigen Weg von A nach C.

b)



Wiederum ausgehend von A findet der Algorithmus den falschen Weg von A nach C, weil hier der Weg über B kürzer wäre. A-C hat das Gewicht 2 und A-B-C hat das Gewicht 0.