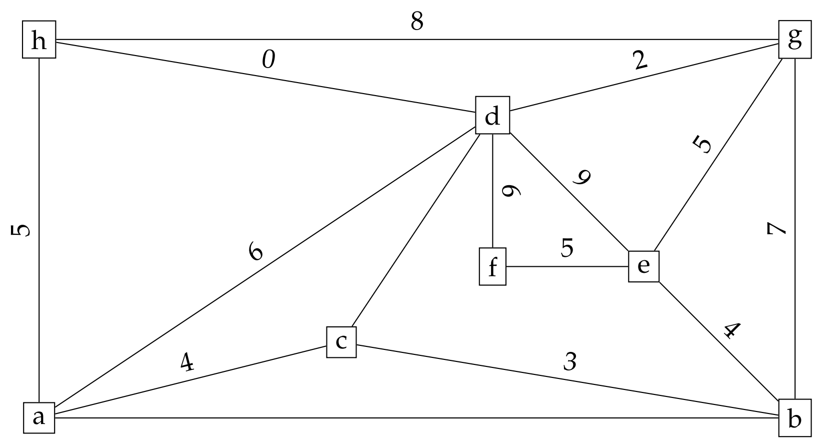
Aufgabe 10:

(a) Berechnen Sie mithilfe des Algorithmus von Prim ausgehend vom Knoten a einen minimalen Spannbaum des ungerichteten Graphen G, der durch folgende Adjazenzmatrix gegeben ist:

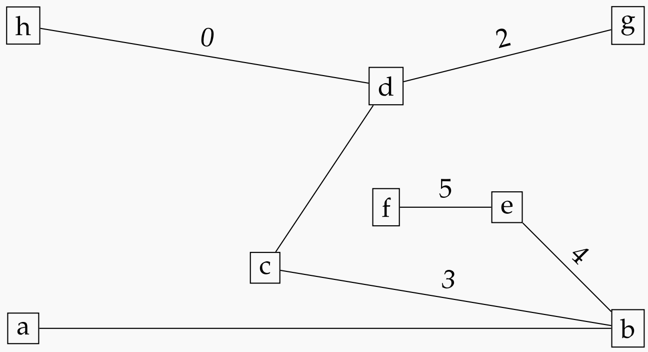
$$
\begin{array}{c c}
 & \begin{array}{cccccccc} a & b & c & d & e & f & g & h \end{array} \\
\begin{array}{c} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{array} &
\begin{pmatrix}
* & 1 & 4 & 6 & - & - & - & 5 \\
1 & * & 3 & - & 4 & - & 7 & - \\
4 & 3 & * & 1 & - & - & - & - \\
6 & - & 1 & * & 9 & 6 & 2 & 0 \\
- & 4 & - & 9 & * & 5 & 5 & - \\
- & - & - & 6 & 5 & * & - & - \\
- & 7 & - & 2 & 5 & - & * & 8 \\
5 & - & - & 0 & - & - & 8 & 1
\end{pmatrix}
\end{array}
$$



Erstellen Sie dazu eine Tabelle mit zwei Spalten und stellen Sie jeden einzelnen Schritt des Verfahrens in einer eigenen Zeile dar. Geben Sie in der ersten Spalte denjenigen Knoten v, der vom Algorithmus als nächstes in den Ergebnisbaum aufgenommen wird (dieser sog. „schwarze" Knoten ist damit fertiggestellt), als Tripel (v, p, δ) mit v als Knotenname, p als aktueller Vorgängerknoten und δ als aktuelle Distanz von v zu p an. Führen Sie in der zweiten Spalte alle anderen vom aktuellen Spannbaum direkt erreichbaren Knoten v (sog. „graue Randknoten") ebenfalls als Tripel (v, p, δ) auf.

Zeichnen Sie anschließend den entstandenen Spannbaum und geben sein Gewicht an.

„schwarze“	„graue“ Randknoten
(a, NULL, ∞)	(b, a, 1) (c, a, 4) (d, a, 6) (h, a, 5)
(b, a, 1)	(c, b, 3) (d, a, 6) (e, b, 4) (g, b, 7) (h, a, 5)
(c, b, 3)	(d, c, 1) (e, b, 4) (g, b, 7) (h, a, 5)
(d, c, 1)	(e, b, 4) (f, d, 6) (g, d, 2) (h, d, 0)
(h, d, 0)	(e, b, 4) (f, d, 6) (g, d, 2)
(g, d, 2)	(e, b, 4) (f, d, 6)
(e, b, 4)	(f, e, 5)
(f, e, 5)	

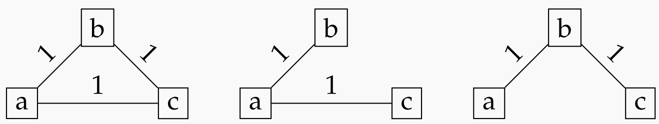


Minimales Kantengewicht: 16

(b) Welche Worst-Case-Laufzeitkomplexität hat der Algorithmus von Prim, wenn die grauen Knoten in einem Heap (= Halde) nach Distanz verwaltet werden? Sei dabei n die Anzahl an Knoten und m die Anzahl an Kanten des Graphen. Eine Begründung ist nicht erforderlich.

$\mathcal{O}(n \cdot \log(n) + m)$

(c) Zeigen Sie durch ein kleines Beispiel, dass ein minimaler Spannbaum eines ungerichteten Graphen nicht immer eindeutig ist.



(d) Skizzieren Sie eine Methode, mit der ein maximaler Spannbaum mit einem beliebigen Algorithmus für minimale Spannbäume berechnet werden kann. In welcher Laufzeitkomplexität kann ein maximaler Spannbaum berechnet werden?

Alle Kantengewichte negieren. In $\mathcal{O}(n \cdot \log(n) + m)$ wie der Algorithmus von Prim.