

Aufgabe

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

$$\begin{array}{c}
 \begin{array}{cccccccc}
 & s & a & b & c & d & e & f & g & h \\
 s & * & - & 3 & - & - & 7 & - & - & - \\
 a & - & * & - & 0 & 8 & - & 11 & - & - \\
 b & 3 & - & * & - & 5 & - & 10 & - & - \\
 c & - & 0 & - & * & - & - & 1 & - & - \\
 d & - & 8 & 5 & - & * & 2 & 3 & - & 6 \\
 e & 7 & - & - & - & 2 & * & - & - & 11 \\
 f & - & 11 & 10 & 1 & 3 & - & * & 7 & - \\
 g & - & - & - & - & - & 7 & * & 4 & - \\
 h & - & - & - & - & 6 & 11 & - & 4 & *
 \end{array}
 \end{array}$$

- (a) 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 Sie sein Gewicht an.

Der Graph muss nicht gezeichnet werden. Der Algorithmus kann auch nur mit der Adjazenzmatrix durchgeführt werden. Möglicherweise geht das Lösen der Aufgabe schneller mit der Matrix von der Hand.

Kompletter Graph

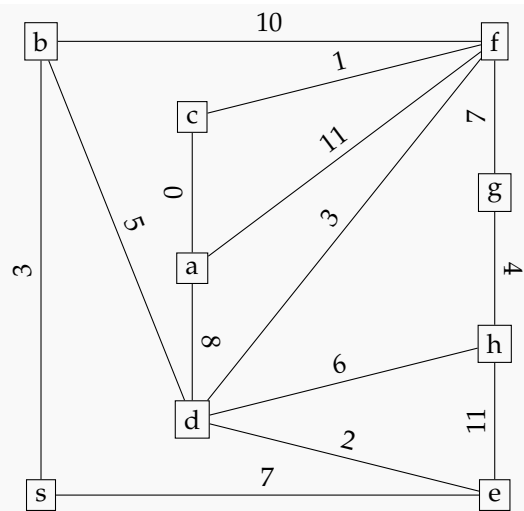
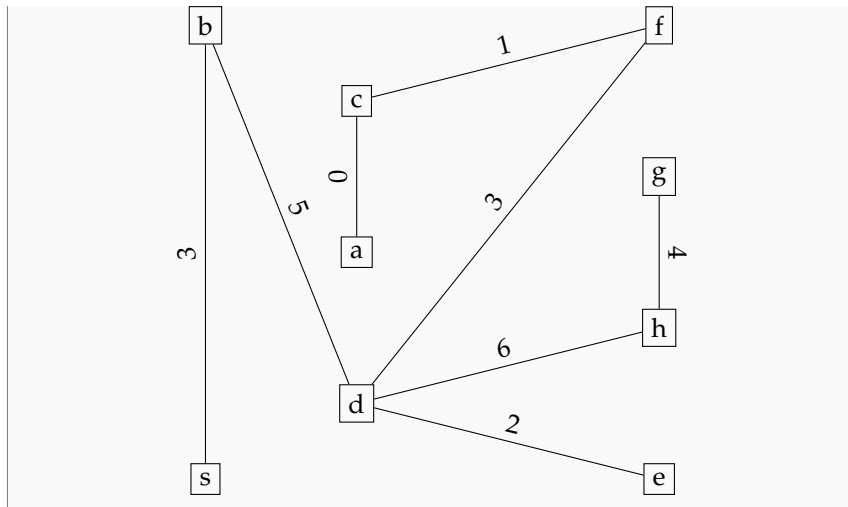


Tabelle schwarz-graue-Knoten

schwarze	graue
(s, null, -)	(b, s, 3); (e, s, 7);
(b, s, 3)	(d, b, 5); (e, s, 7); (f, b, 10);
(d, b, 5)	(a, d, 8); (e, d, 2); (f, d, 3); (h, d, 6);
(e, d, 2)	(a, d, 8); (f, d, 3); (h, d, 6);
(f, d, 3)	(a, d, 8); (c, f, 1); (g, f, 7); (h, d, 6);
(c, f, 1)	(a, c, 0); (g, f, 7); (h, d, 6);
(a, c, 0)	(g, f, 7); (h, d, 6);
(h, d, 6)	(g, h, 4);
(g, h, 4)	

Gewicht des minimalen Spannbaums: 24

Minimaler Spannbaum



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

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

- (c) Beschreiben Sie kurz die Idee des alternativen Ansatzes zur Berechnung eines minimalen Spannbaumes von Kruskal.

Kruskal wählt nicht die kürzeste an einen Teilgraphen anschließende Kante, sondern global die kürzeste verbliebene aller Kanten, die keinen Zyklus bildet, ohne dass diese mit dem Teilgraph verbunden sein muss.