

---

**Algorithm 1** Kruskal's Algorithm

---

**INPUT:** Ungerichteter, zusammenhängender Graph  $G = (V, E)$  mit Kantengewichten  $c : E \rightarrow \mathbb{R}$

**OUTPUT:** Ein minimaler Spannbaum  $T$ .

- 1: Sortiere die Kanten:  $c(e_1) \leq c(e_2) \leq \dots \leq c(e_m)$
  - 2: Setze  $T := (V(G), \emptyset)$
  - 3: **for**  $i := 1$  **to**  $m$  **do**
  - 4:     **if**  $T + e_i$  enthält keinen Kreis **then**
  - 5:         Setze  $T := T + e_i$
  - 6:     **end if**
  - 7: **end for**
- 

---

**Algorithm 2** Prim's Algorithm

---

**INPUT:** Ungerichteter, zusammenhängender Graph  $G = (V, E)$  mit Kantengewichten  $c : E \rightarrow \mathbb{R}$

**OUTPUT:** Ein minimaler Spannbaum  $T$ .

- 1: Wähle einen beliebigen Knoten  $v \in V(G)$  und setze  $T := (\{v\}, \emptyset)$ .
  - 2: **while**  $V(T) \neq V(G)$  **do**
  - 3:     Wähle die billigste Kante  $e = (x, y) \in E(G)$  mit  $x \in V(T)$ ,  $y \notin V(T)$  und setze  $T := T + e$ .
  - 4: **end while**
-