Algorithm 1 Kruskal's Algorithm

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

OUTPUT: Ein minimaler Spannbaum T.

Sortiere die Kanten: $c(e_1) \leq c(e_2) \leq \ldots \leq c(e_m)$
Setze $T := (V(G), \emptyset)$
for $i := 1$ **to** m **do**
    if $T + e_i$ enthält keinen Kreis **then**
        Setze $T := T + e_i$
    end if
end for

Algorithm 2 Prim's Algorithm

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

OUTPUT: Ein minimaler Spannbaum T.

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