

2.5 Eine Haldenimplementierung des Prim-Verfahrens

Prim Algorithmus mit Haldenimplementierung

1. Input: ungerichteter zusammenhängender gewichtete Graph $G = (V, E)$ mit Kantengewichten $w(i, j) \forall (i, j) \in E$.
2. Initialisierung:
 - Halde h ist leer,
 - $W(j) := \infty$, für $j = 2, 3, \dots, n$.
 - Setze den Baum $T := \emptyset$ und die Schnitt-definierende Teilmenge $X = \{1\}$
3. Wähle $j^* = 1$.
4. Untersuche alle Nachbarn k , $k \notin X$, von j^* in G :
 - Falls $w(j^*, k) < W(k)$, setze
 - $W(k) := w(j^*, k)$,
 - $N(k) := -j^*$.
 - Falls $k \notin h$, füge den Knoten k mit Schlüssel $W(k)$ zur Halde h hinzu.
 - Falls $k \in h$, ersetze den Schlüssel von k durch $W(k)$ und schiebe diesen Knoten in der Halde nach oben durch.
5. Falls $h = \emptyset$, gehe zu Schritt 7.
6. Bestimme j^* als den Knoten der Halde mit dem kleinsten Schlüssel und streiche diesen Knoten in h . Setze $X := X \cup \{j^*\}$, $T := T \cup \{(N(j^*), j^*)\}$. Gehe zu Schritt 7
7. Terminiere. Output den minimalen spannenden Baum T .