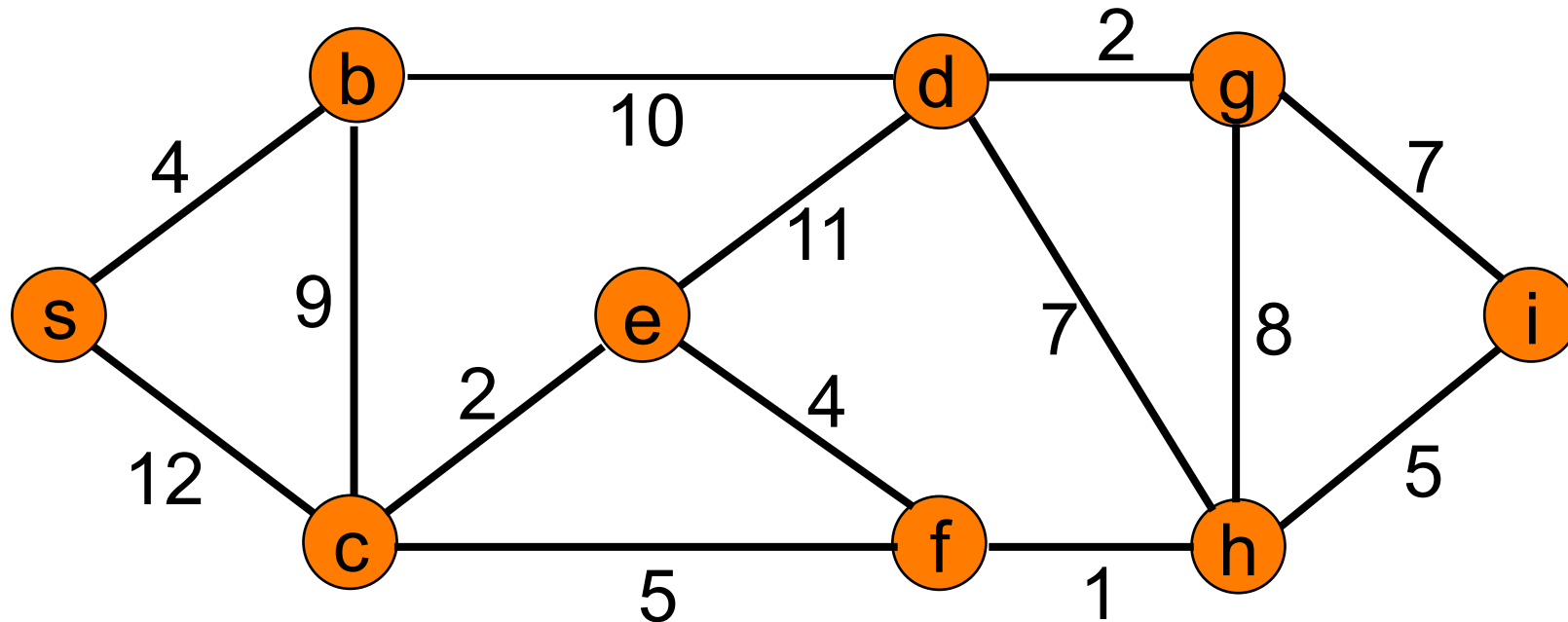
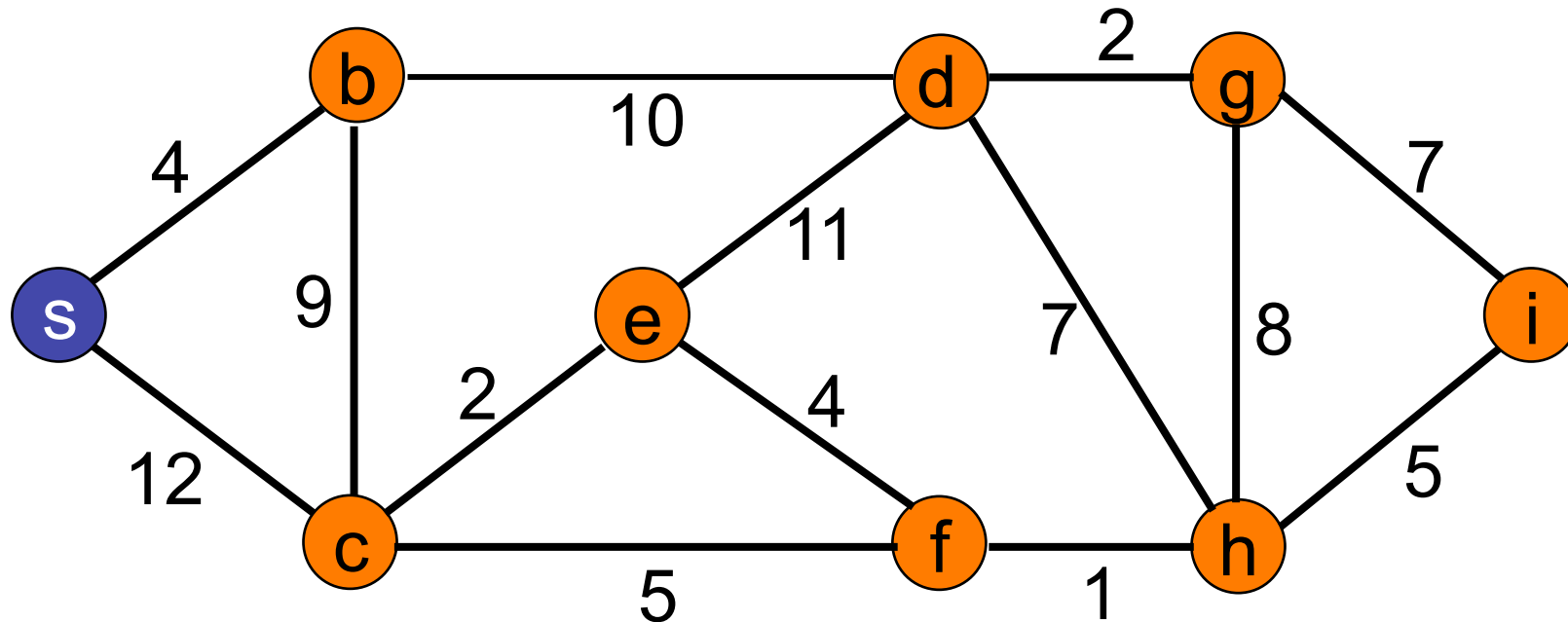


# Algorithmus von Prim



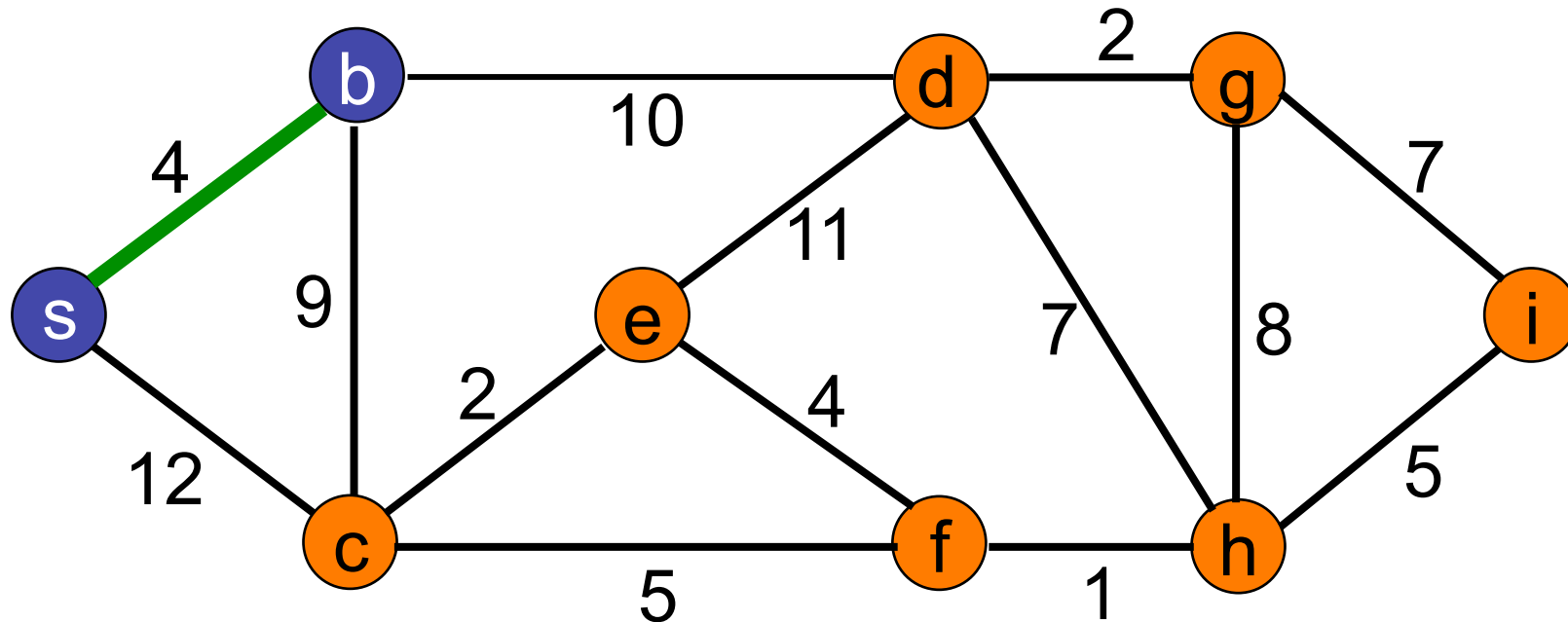
Knotenmenge  $S$ , Kantenmenge  $T$ :  
In Teilbaum  $T_i$

# Algorithmus von Prim



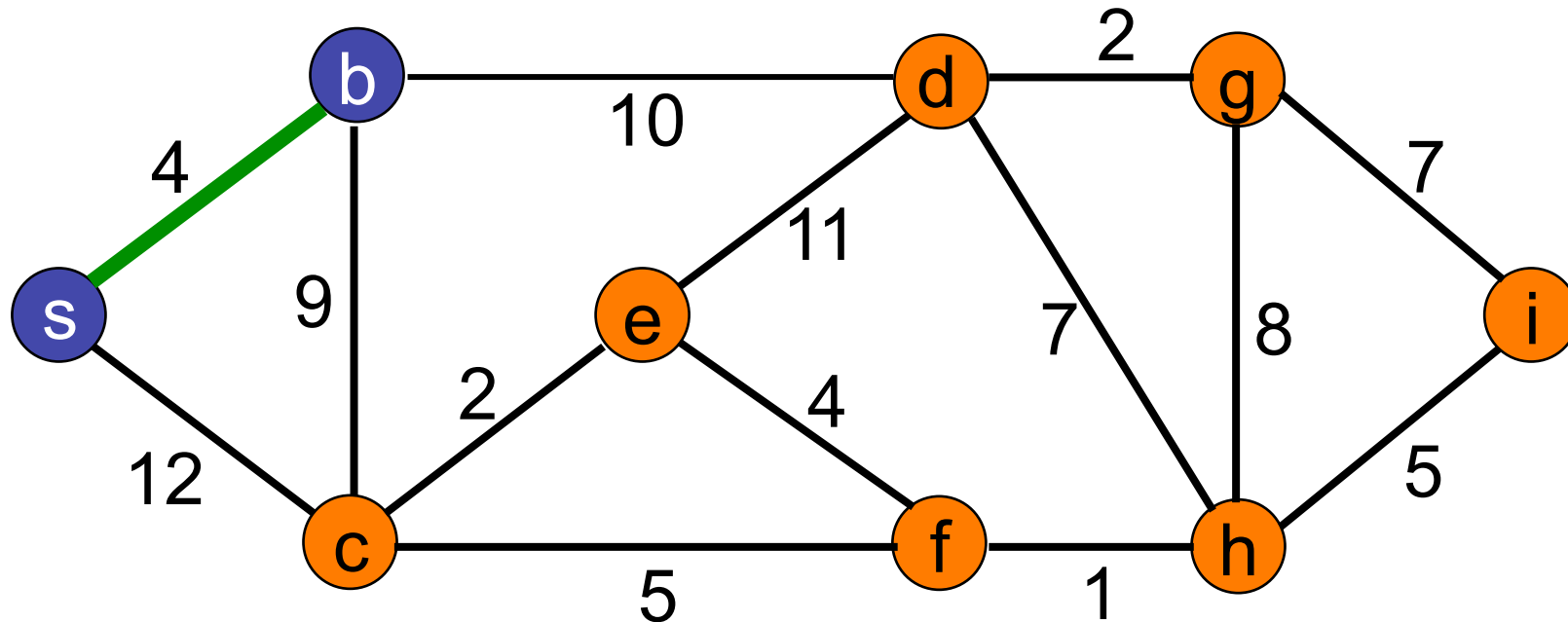
Knotenmenge S, Kantenmenge T:  
In Teilbaum  $T_i$

# Algorithmus von Prim



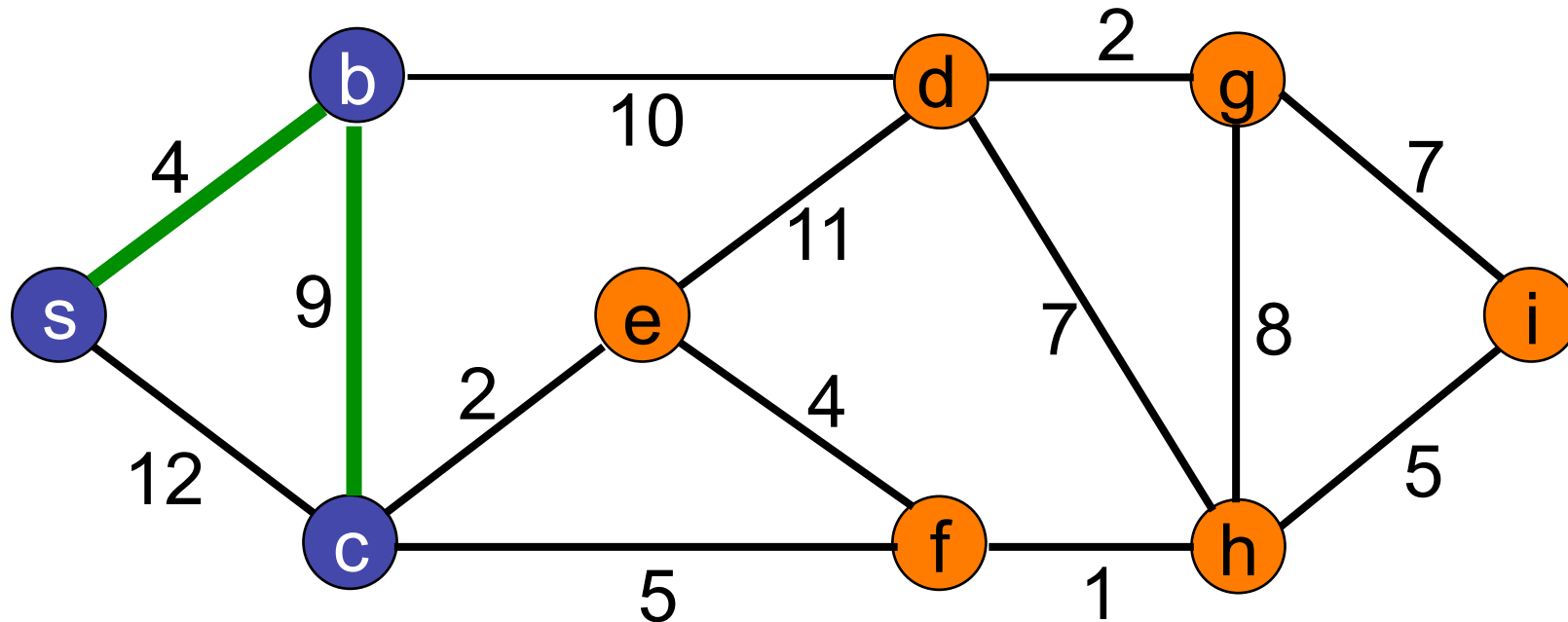
Knotenmenge S, Kantenmenge T:  
In Teilbaum  $T_i$

# Algorithmus von Prim



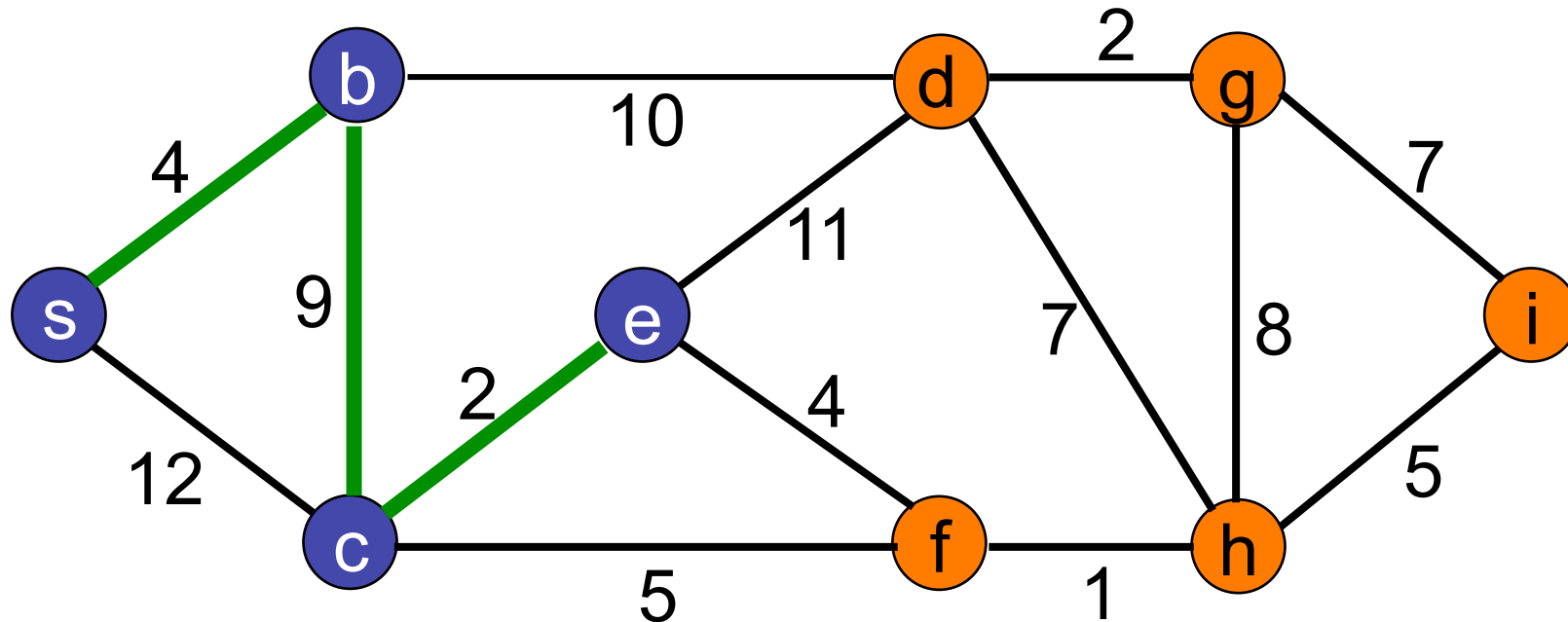
Knotenmenge S, Kantenmenge T:  
In Teilbaum  $T_i$

# Algorithmus von Prim



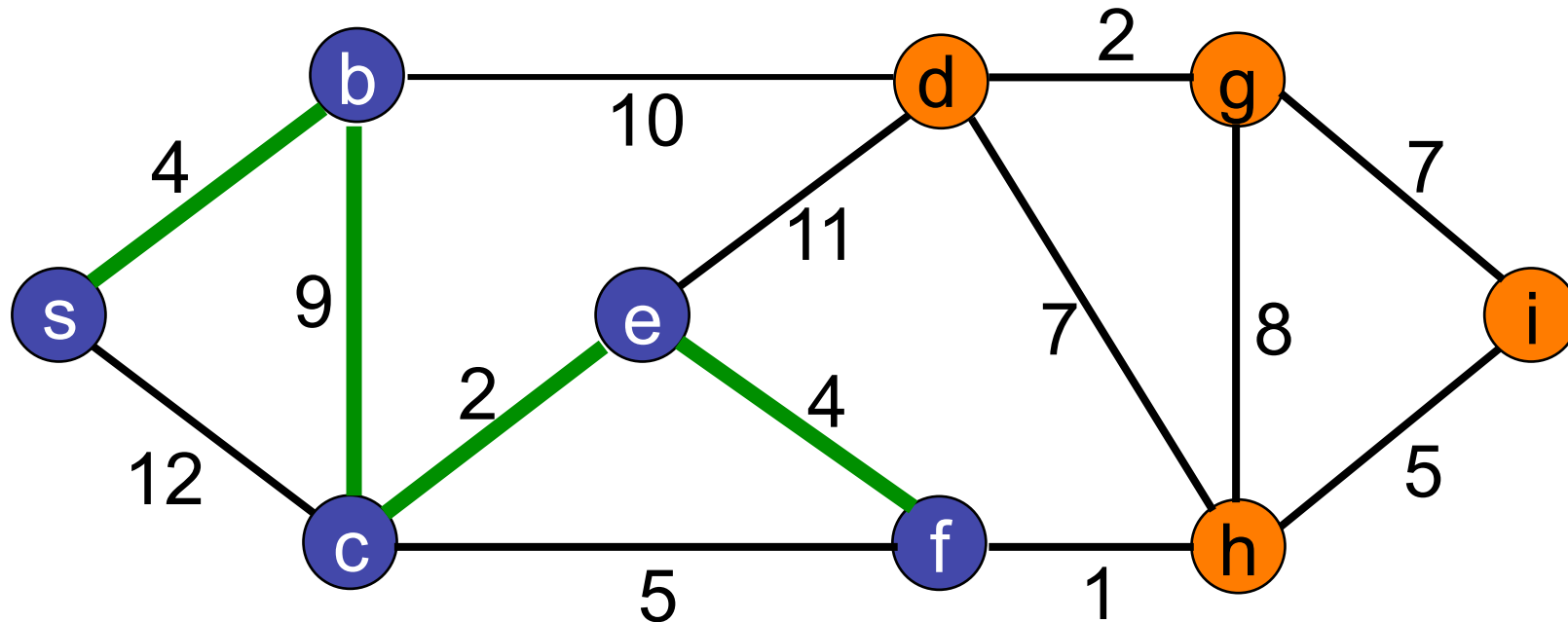
Knotenmenge S, Kantenmenge T:  
In Teilbaum  $T_i$

# Algorithmus von Prim



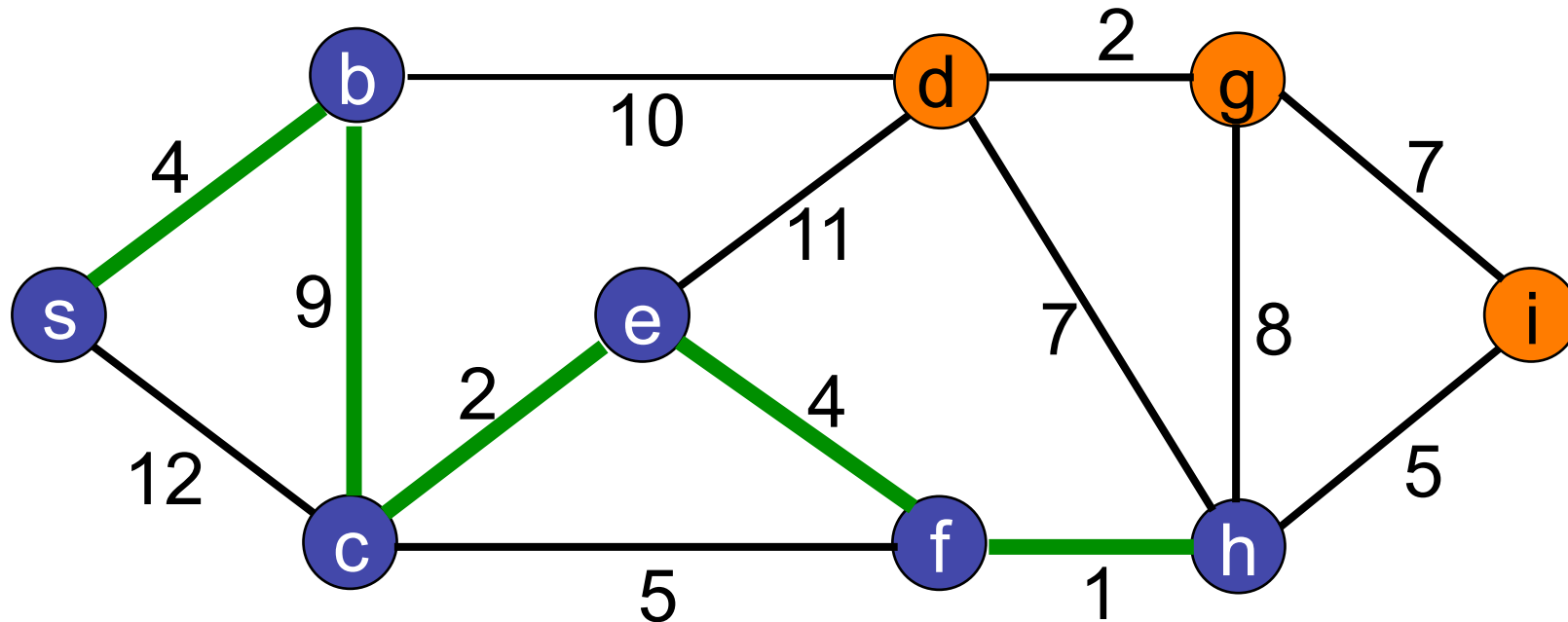
Knotenmenge  $S$ , Kantenmenge  $T$ :  
In Teilbaum  $T_i$

# Algorithmus von Prim



Knotenmenge  $S$ , Kantenmenge  $T$ :  
In Teilbaum  $T_i$

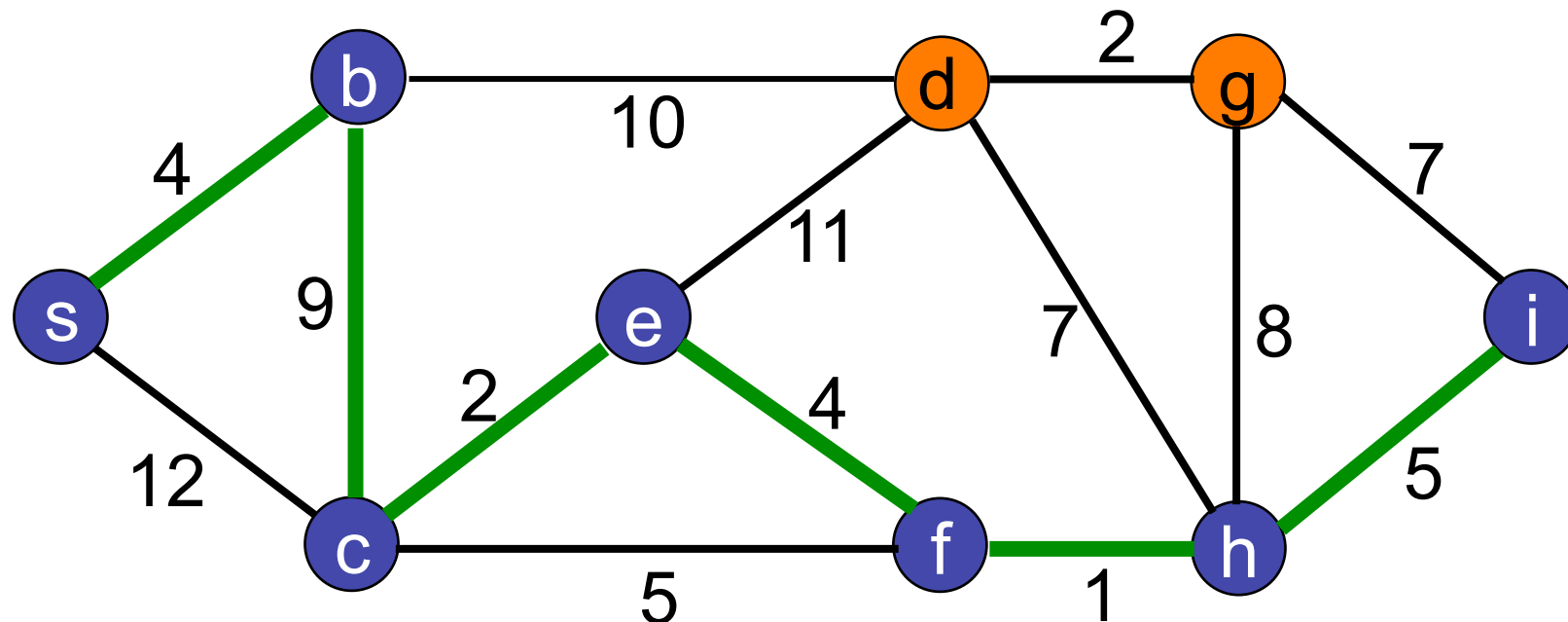
# Algorithmus von Prim



Knotenmenge  $S$ , Kantenmenge  $T$ :  
In Teilbaum  $T_i$

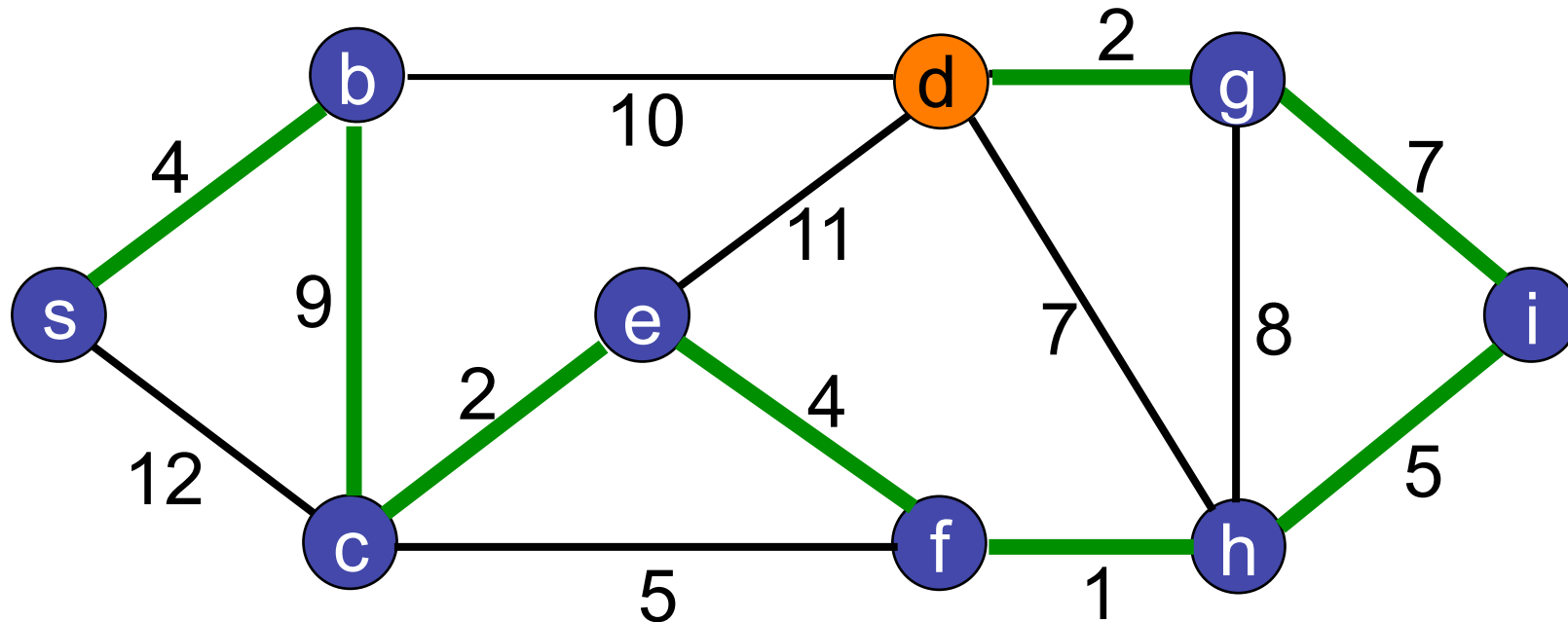


# Algorithmus von Prim



Knotenmenge  $S$ , Kantenmenge  $T$ :  
In Teilbaum  $T_i$

# Algorithmus von Prim



Knotenmenge S, Kantenmenge T:  
In Teilbaum T<sub>i</sub>

# Realisierung von Prim

- Testen auf Kreise (Union-Find Partition) wie bei Kruskal nicht nötig
- Kein Sortieren nötig
- Auswahl leichtester Kante nötig  
⇒ Priority Queue
- Wir speichern nicht die Kanten selbst, sondern die Knoten  $v \in V \setminus S$ . Priorität ist das Gewicht der Kante von  $S$  nach  $v$