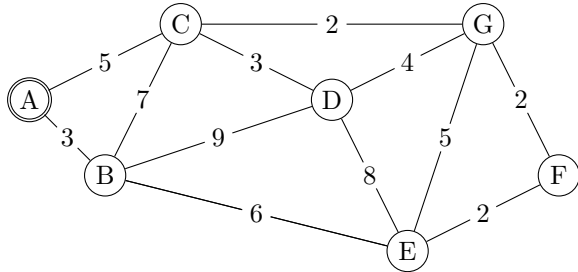


# Der Algorithmus von Dijkstra

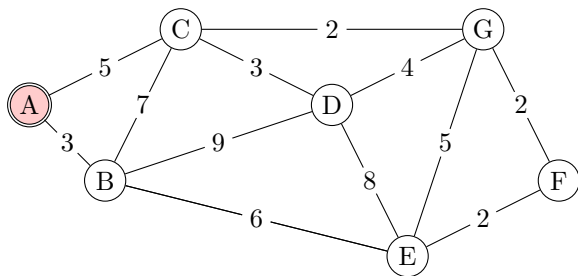
## Die Ausgangslage

Gegeben: Ein kantengewichteter Graph  $G$  und ein Startknoten  $A$

Gesucht: Den kürzesten Weg zu jedem Knoten im Graph



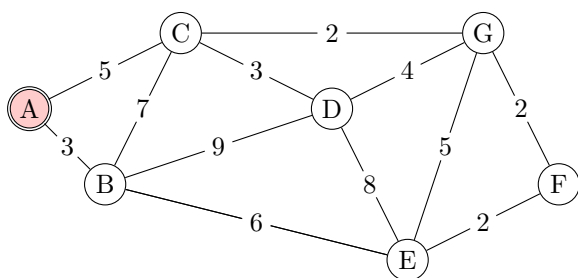
## Schritt 1



Der Startknoten (A) ist der erste Arbeitsknoten (rot).

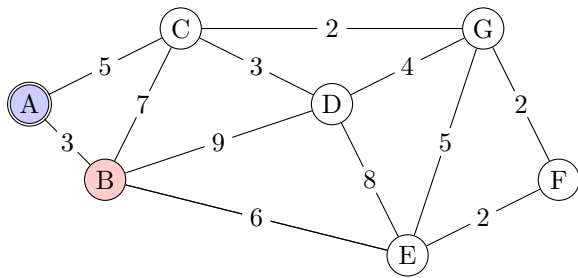
Initialisiere jeden Knoten mit der Gesamtdistanz vom Startknoten. Abgesehen vom Startknoten wird dafür  $\infty$  gewählt. *Grund:* Falls der Graph isolierte Knoten hat, ist dies ein sinnvoller Distanzwert.

## Schritt 2



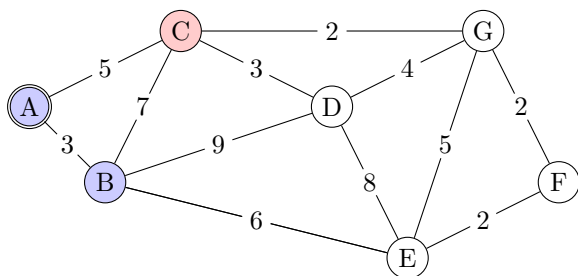
Gehe vom Arbeitsknoten A (rot) zu jedem seiner Nachbarn und untersuche, ob die Distanz zum Zielknoten „relaxiert“ (entspannt, unterschritten) werden kann. Falls ja, notiere den Arbeitsknoten und die neue Gesamtdistanz.

### Schritt 3



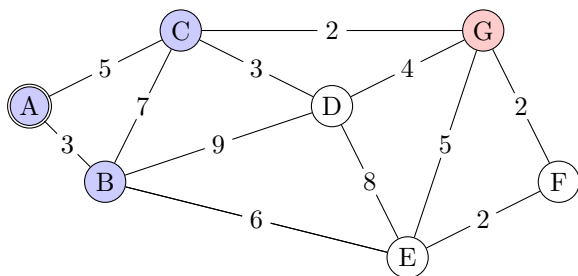
Kennzeichne den letzten Arbeitsknoten A als verarbeitet (blau) und wähle aus den übrigen Knoten denjenigen mit der kleinsten Gesamtdistanz vom Startknoten aus. Dies ist B. Relaxiere die Kanten zu allen Nachbarn des Arbeitsknotens wie im letzten Schritt.

### Schritt 4



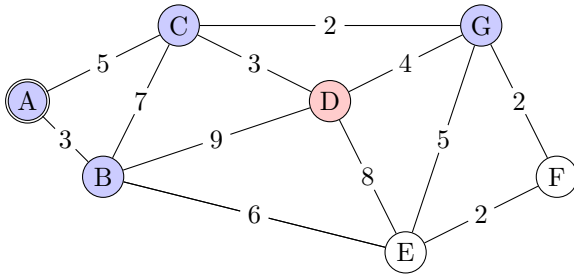
Kennzeichne den letzten Arbeitsknoten B als verarbeitet (blau) und wähle aus den übrigen Knoten denjenigen mit der kleinsten Gesamtdistanz vom Startknoten aus. Dies ist C (rot). Relaxiere die Kanten zu allen Nachbarn des Arbeitsknotens wie im letzten Schritt.

### Schritt 5



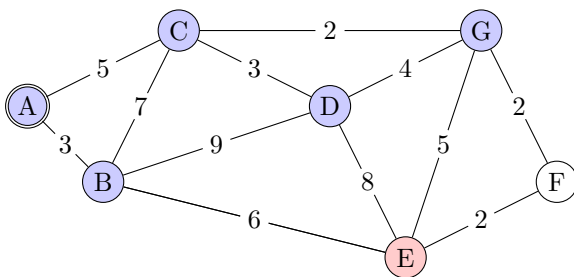
Kennzeichne den letzten Arbeitsknoten C als verarbeitet (blau) und wähle aus den übrigen Knoten denjenigen mit der kleinsten Gesamtdistanz vom Startknoten aus. Dies ist G (rot). Relaxiere wie im letzten Schritt die Kanten zu allen Nachbarn des Arbeitsknotens.

### Schritt 6



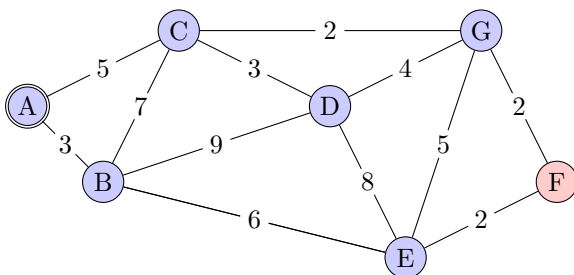
Kennzeichne den letzten Arbeitsknoten (G) als verarbeitet (blau) und wähle aus den übrigen Knoten denjenigen mit der kleinsten Gesamtentfernung vom Startknoten aus. Dies ist D (rot). Relaxiere wie im letzten Schritt die Kanten zu allen Nachbarn des Arbeitsknotens.

### Schritt 7



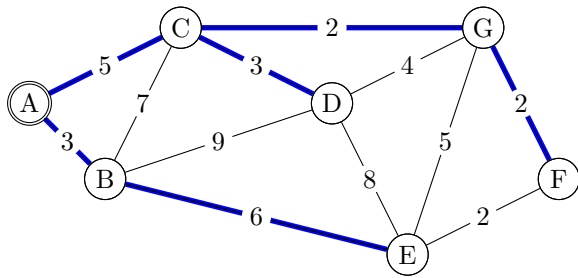
Kennzeichne den letzten Arbeitsknoten (D) als verarbeitet (blau) und wähle aus den übrigen Knoten denjenigen mit der kleinsten Gesamtentfernung vom Startknoten aus. Dies ist E (rot). Man hätte aber auch F wählen können. Relaxiere wie im letzten Schritt die Kanten zu allen Nachbarn des Arbeitsknotens.

### Schritt 8



Sobald der letzte vom Startknoten erreichbare Knoten verarbeitet, d.h. relaxiert wurde, endet der Algorithmus von Dijkstra

## Der Spannbaum



Verfolgt man gemäss den Knotenbeschriftungen die Pfade von den einzelnen Knoten zurück, erhält man einen *Spannbaum*. Dieser Baum enthält jeden Knoten des Graphen und die Summe der Kantengewichte vom Wurzelknoten (A) bis zum jeweiligen Knoten ist minimal. *Beispiel*: Der kürzteste Weg von (A) nach (F) geht über (C) und (G).