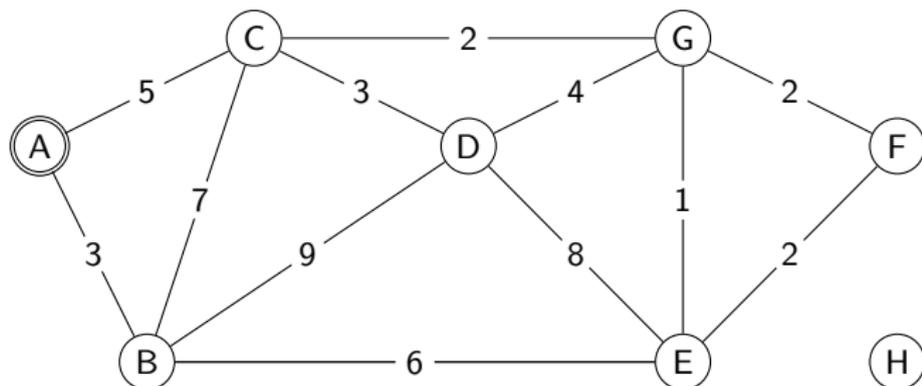


Der Algorithmus von Dijkstra

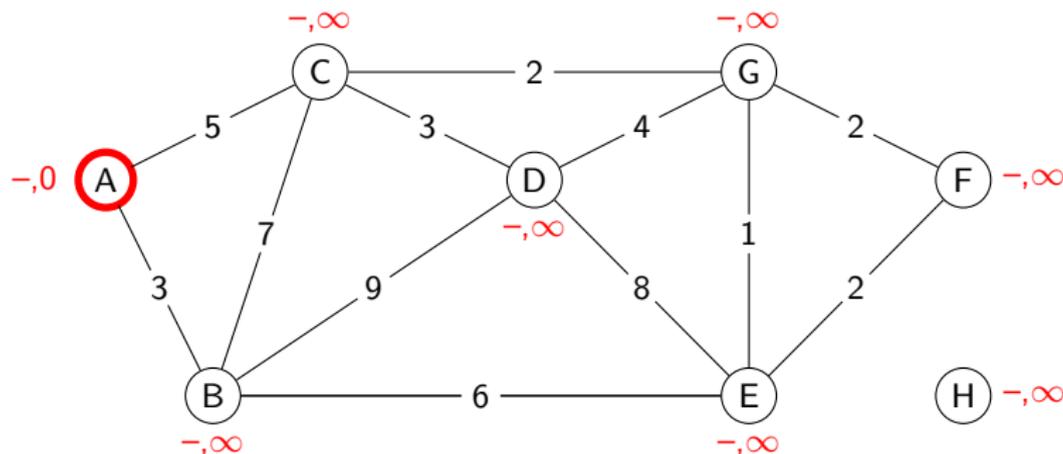
Die Ausgangslage

Gegeben: Kantengewichteter Graph G und Startknoten A

Gesucht: Kürzester Weg von A zu jedem Knoten im Graph

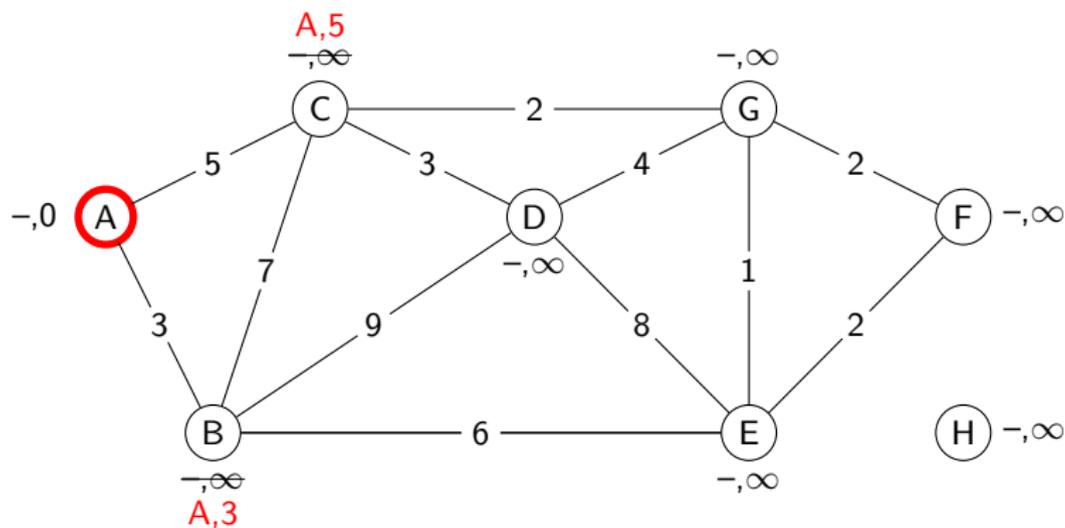


Schritt 1



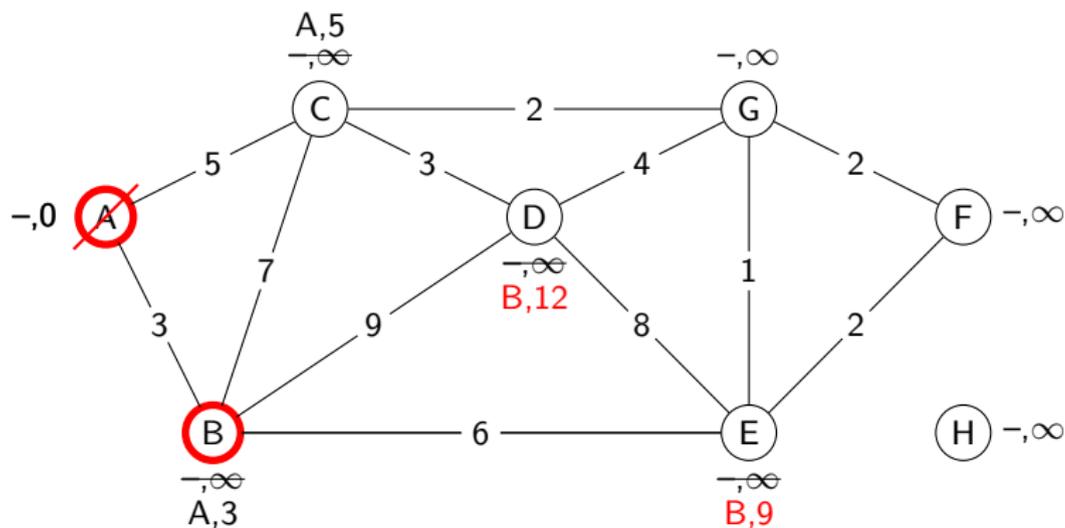
Mache den Startknoten A zum ersten Arbeitsknoten (rot) und initialisiere jeden von A verschiedenen Knoten provisorisch mit einer leeren Herkunftsangabe (-) und der Distanz $∞$, was sinnvoll ist, wenn ein Knoten von A aus nicht erreichbar ist.

Schritt 2



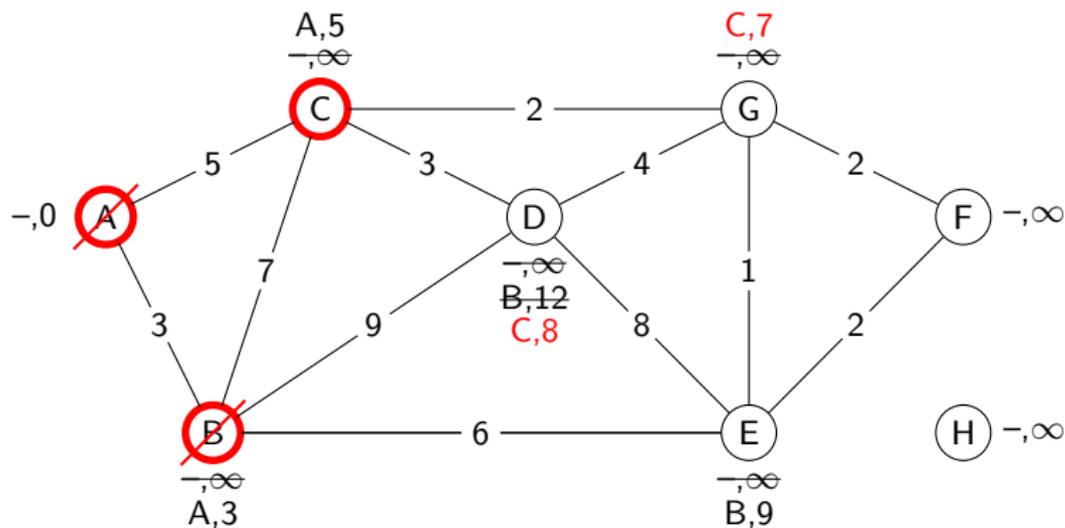
Gehe vom Arbeitsknoten A (rot) zu jedem Nachbarn, der noch nicht Arbeitsknoten war und untersuche, ob die Distanz zum Zielknoten verkleinert (*relaxiert*) werden kann. Falls ja, ersetze den alten Herkunftsknoten und die alte Distanz zum Startknoten.

Schritt 3



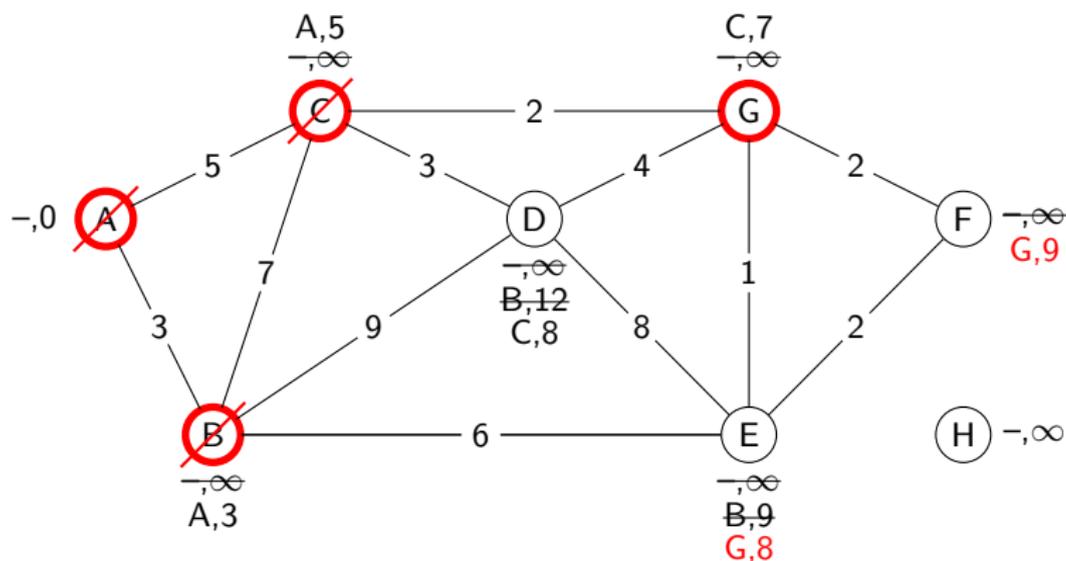
Markiere den Arbeitsknoten A als erledigt und wähle unter den unerledigten Knoten den mit der kleinsten Gesamtdistanz vom Startknoten (\rightarrow B). Wiederhole Schritt 2 mit Knoten B.

Schritt 4



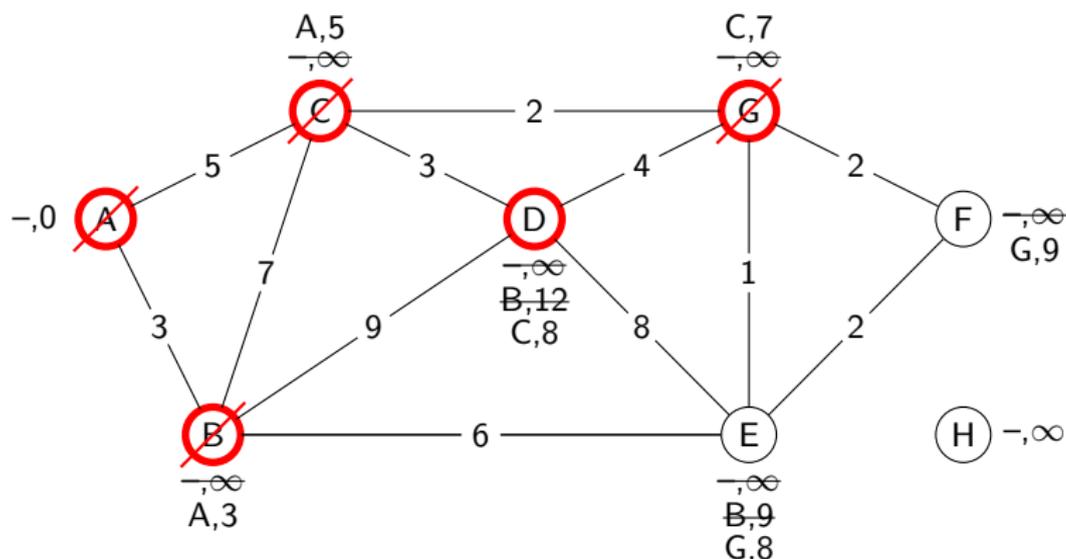
Markiere Arbeitsknoten B als erledigt und wähle unter den unerledigten Knoten den mit der kleinsten Gesamtdistanz vom Startknoten (\rightarrow C). Wiederhole Schritt 2 mit Knoten C.

Schritt 5



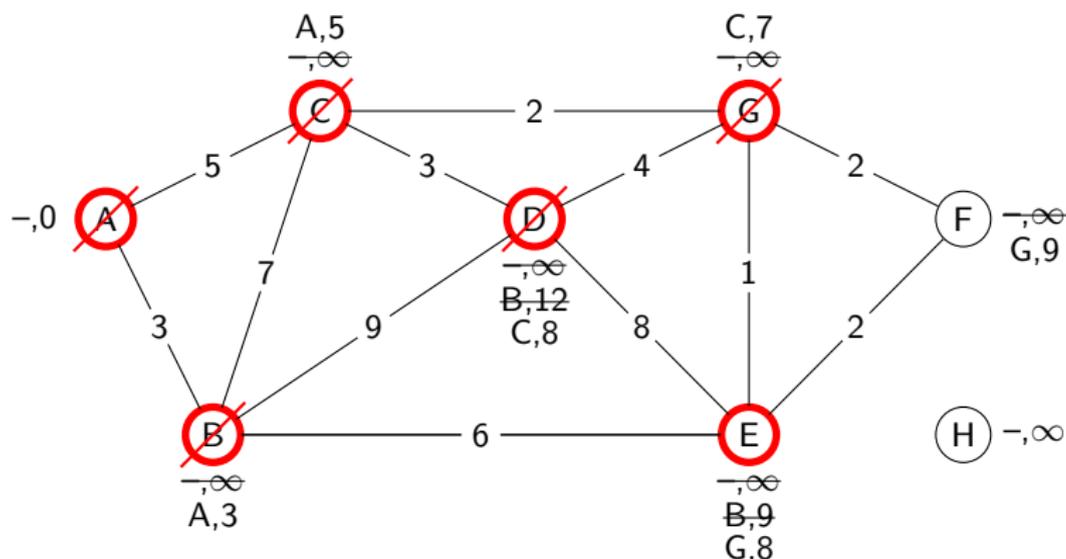
Markiere Arbeitsknoten C als erledigt und wähle unter den unerledigten Knoten den mit der kleinsten Gesamtdistanz vom Startknoten (\rightarrow G). Wiederhole Schritt 2 mit Knoten G.

Schritt 6



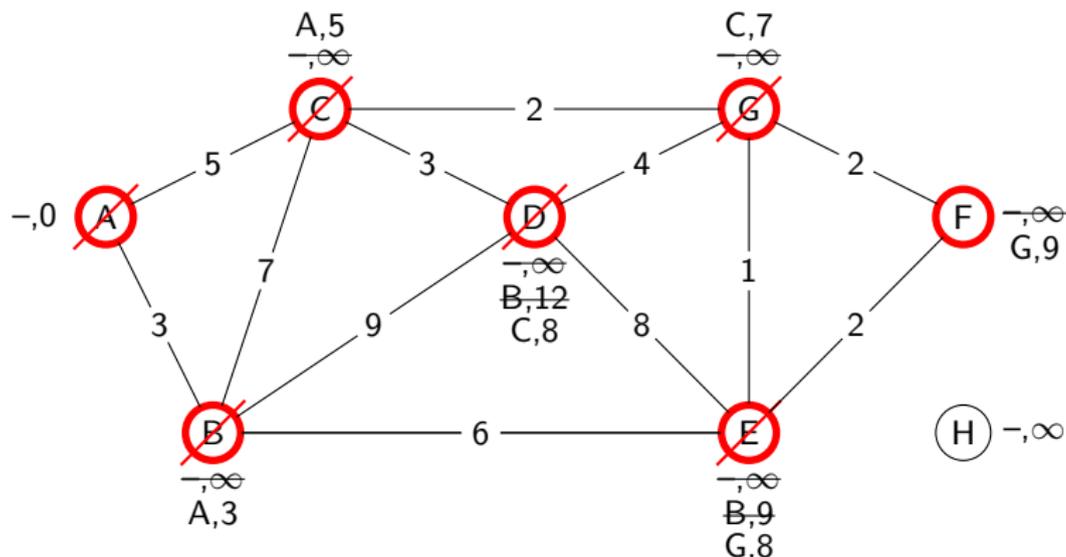
Markiere Arbeitsknoten C als erledigt und wähle unter den unerledigten Knoten den mit der kleinsten Gesamtentfernung vom Startknoten. Da es zwei solche Knoten (D und E) gibt, wählen wir willkürlich Knoten D und wiederholen damit Schritt 2.

Schritt 7



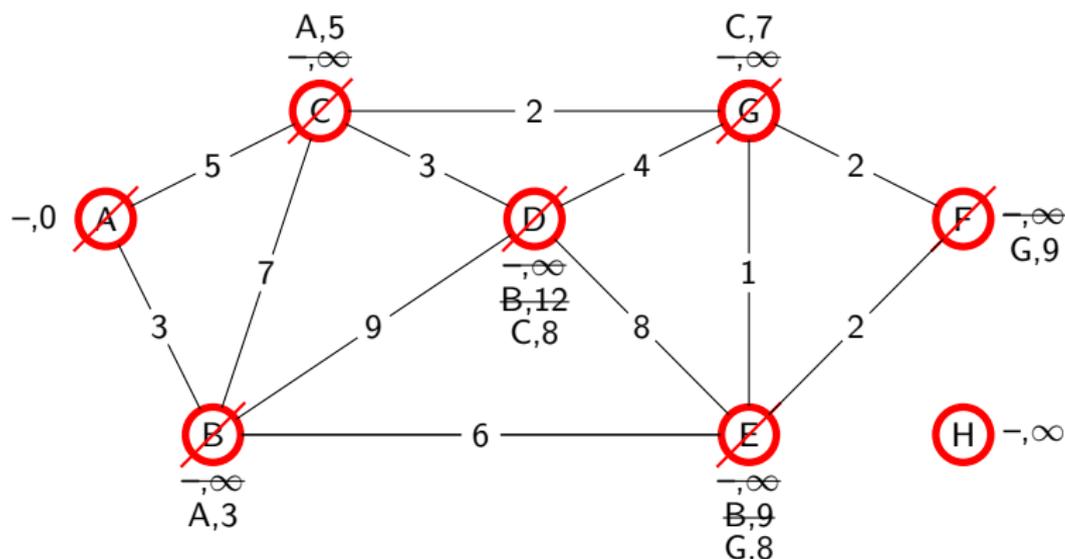
Markiere Arbeitsknoten D als erledigt und wähle unter den unerledigten Knoten den mit der kleinsten Gesamtentfernung vom Startknoten (\rightarrow E). Wiederhole damit Schritt 2.

Schritt 8



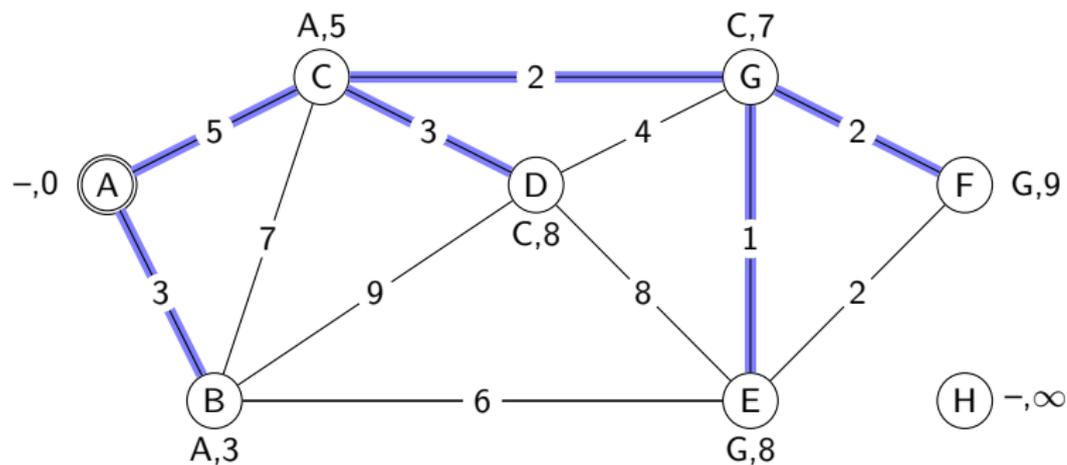
Markiere Arbeitsknoten E als erledigt und wähle unter den noch unerledigten Knoten den mit der kleinsten Gesamtdistanz vom Startknoten (\rightarrow F). Wiederhole damit Schritt 2.

Schritt 9



Markiere Arbeitsknoten F als erledigt und wähle unter den noch unerledigten Knoten den mit der kleinsten Gesamtentfernung vom Startknoten (\rightarrow H). Da es keine weiteren unerledigten Knoten mehr gibt endet hier der Algorithmus von Dijkstra.

Der Spannbaum



Geht man von einem Knoten v_1 zu seinem Vorgängerknoten v_2 und von dort wieder zu dessen Vorgängerknoten v_3 usw., erhält man einen kürzesten Pfad zum Startknoten A. Zusammen bilden diese Pfade einen **Spannbaum**, der im Allgemeinen aber nicht minimal sein muss, d. h. bei dem die Summe aller Kanten minimal ist.

Beschreibung des Algorithmus von Dijkstra

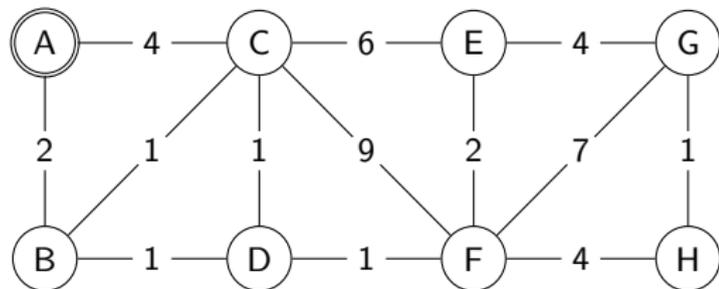
Input: Graph G mit Kantengewichten, Startknoten S

1. *Initialisierung:* Weise jedem Knoten provisorisch seinen *Vorgänger* ($-$) und die Distanz zum Startknoten (0 bzw. ∞) zu. Markiere alle Knoten als unbesucht.
2. Solange es noch unbesuchte (unerledigte) Knoten gibt:
 - 2.1 Wähle darunter den Knoten v mit der kürzesten Distanz zum Startknoten und markiere ihn als besucht. Falls es mehrere solche Knoten gibt, wähle einen beliebigen aus.
 - 2.2 Berechne für alle Nachbarn w von v die Distanz $d(S, w) = d(S, v) + d(v, w)$.
 - 2.3 *Relaxation:* Ist diese Distanz kleiner als die, welche vorher im Knoten w gespeichert wurde, ersetze die alte Distanz durch die neue und speichere den neuen Vorgänger v . Andernfalls ändere nichts.

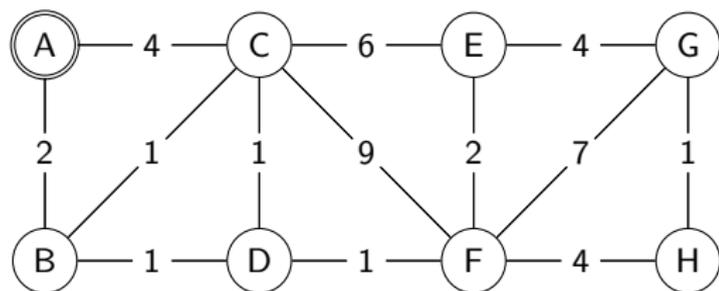
Output: Alle Knoten von G mit ihren Vorgängern und Distanzen

Aufgabe 1

Bestimme die kürzesten Wege vom Startknoten zu allen übrigen Knoten mit dem Algorithmus von Dijkstra. Protokolliere den Lösungsweg, indem du alle Knoten initialisierst und mit dem Algorithmus aktualisierst.

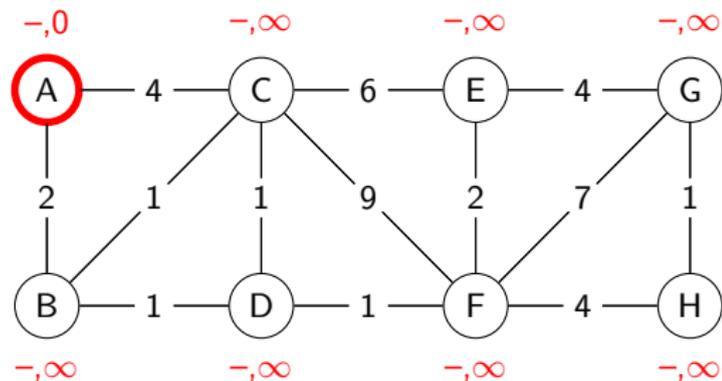


Aufgabe 1



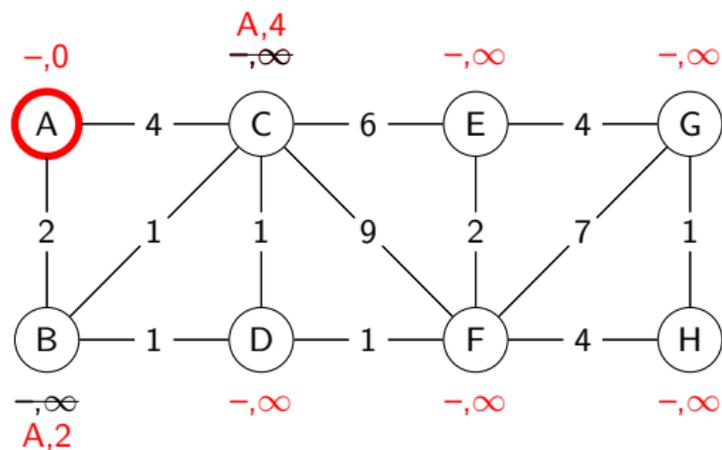
Reihenfolge der Arbeitsknoten:

Aufgabe 1



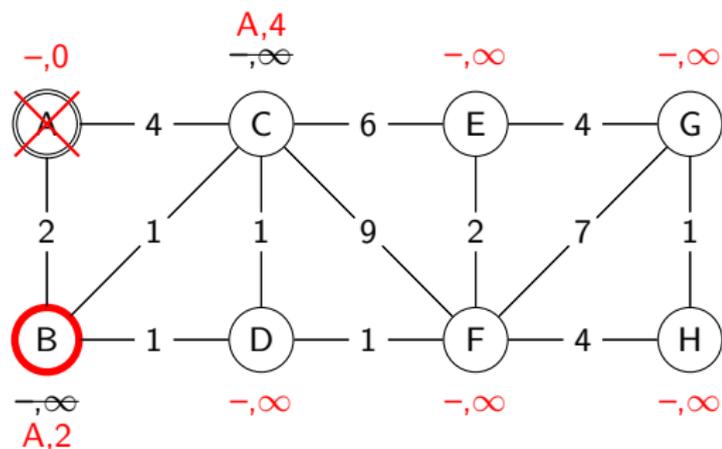
Reihenfolge der Arbeitsknoten: A

Aufgabe 1



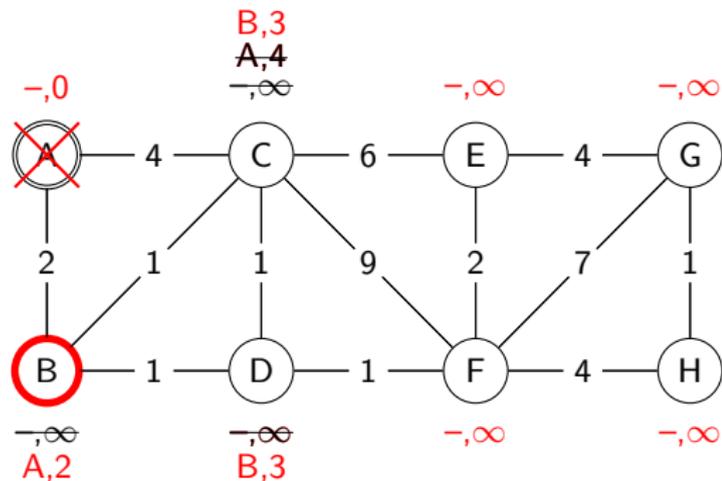
Reihenfolge der Arbeitsknoten: A

Aufgabe 1



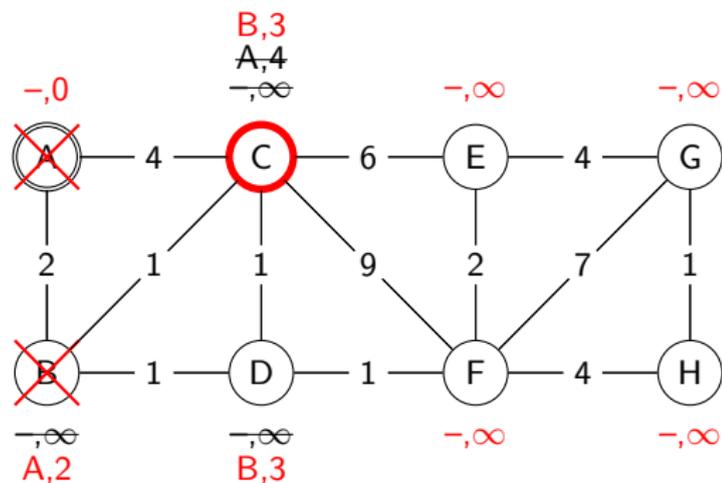
Reihenfolge der Arbeitsknoten: A, B

Aufgabe 1



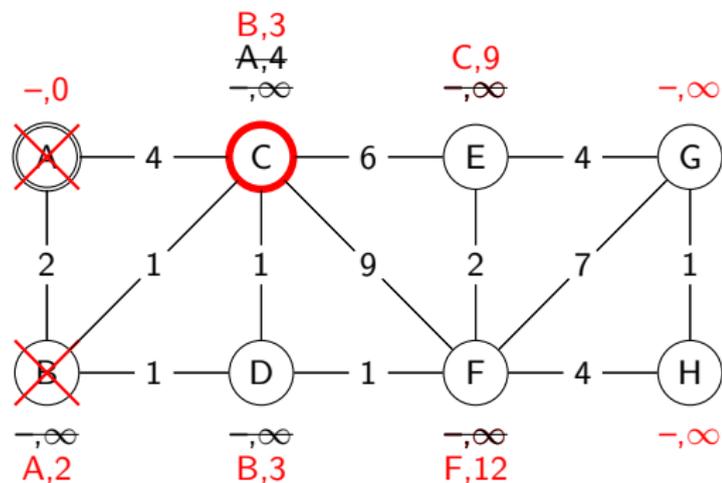
Reihenfolge der Arbeitsknoten: A, B

Aufgabe 1



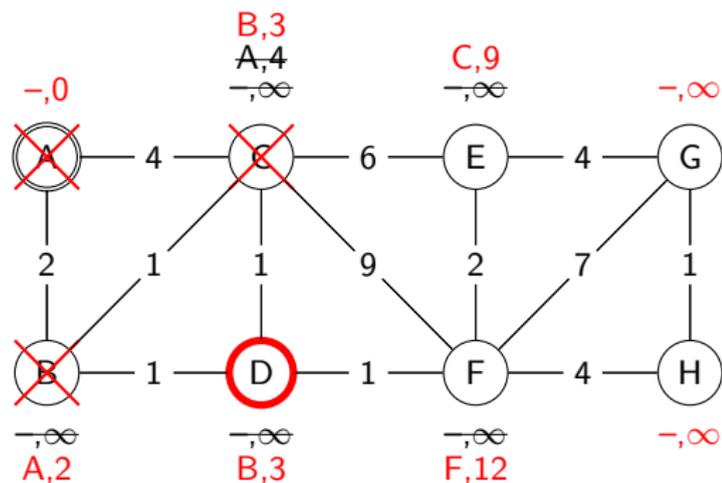
Reihenfolge der Arbeitsknoten: A, B, C

Aufgabe 1



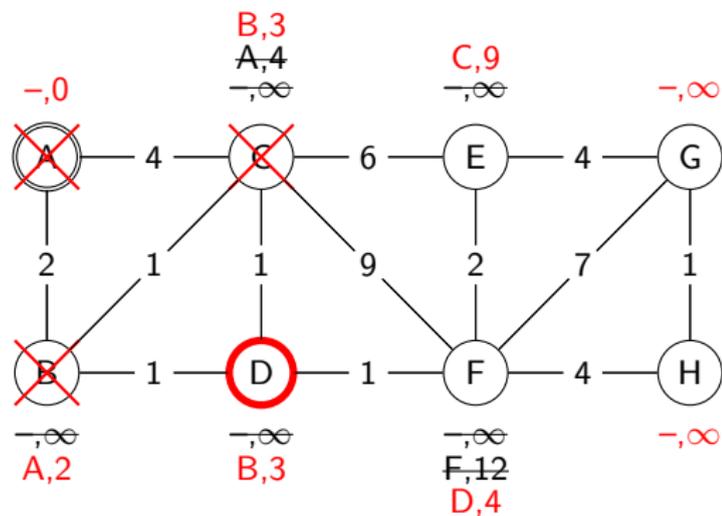
Reihenfolge der Arbeitsknoten: A, B, C

Aufgabe 1



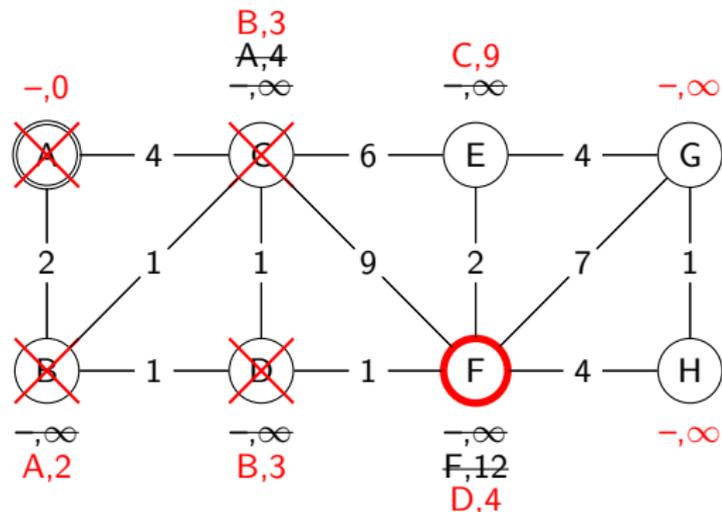
Reihenfolge der Arbeitsknoten: A, B, C, D

Aufgabe 1



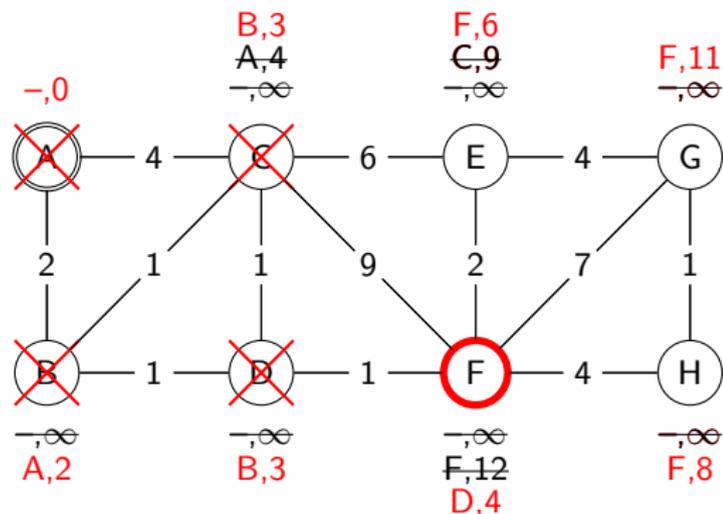
Reihenfolge der Arbeitsknoten: A, B, C, D

Aufgabe 1



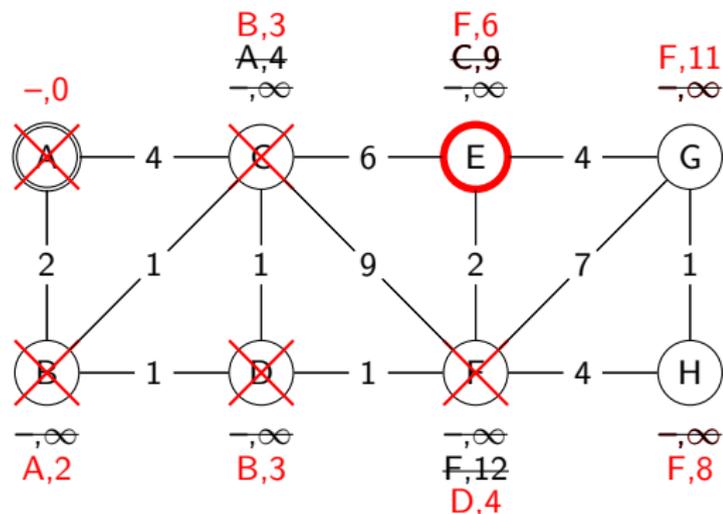
Reihenfolge der Arbeitsknoten: A, B, C, D, F

Aufgabe 1



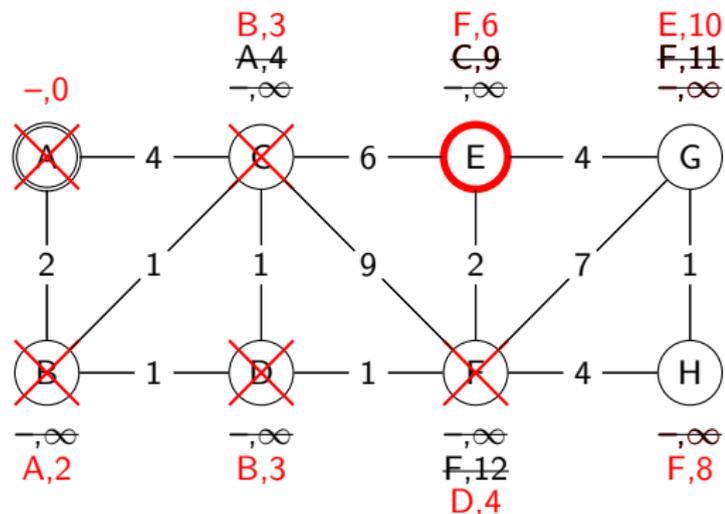
Reihenfolge der Arbeitsknoten: A, B, C, D, F

Aufgabe 1



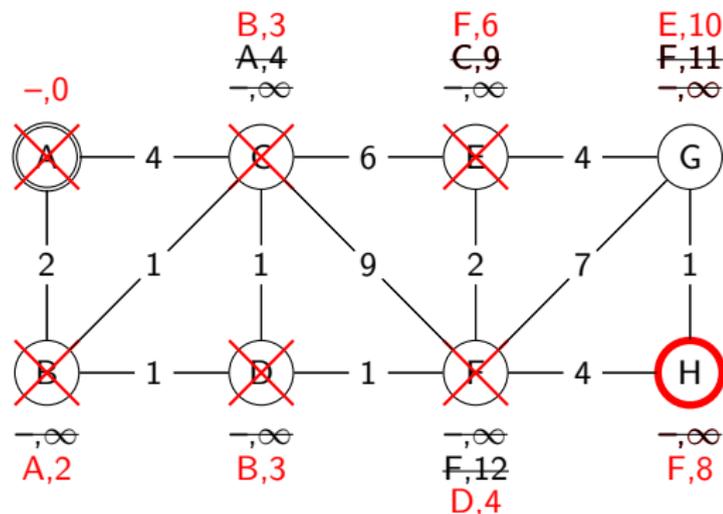
Reihenfolge der Arbeitsknoten: A, B, C, D, F, E

Aufgabe 1



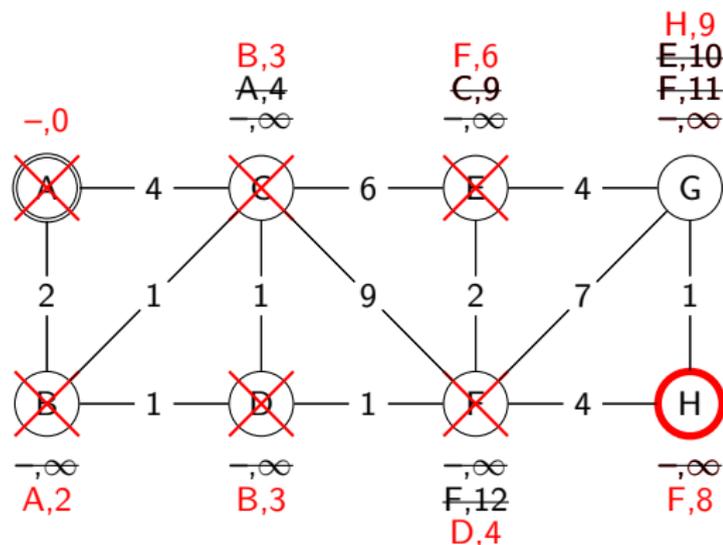
Reihenfolge der Arbeitsknoten: A, B, C, D, F, E

Aufgabe 1



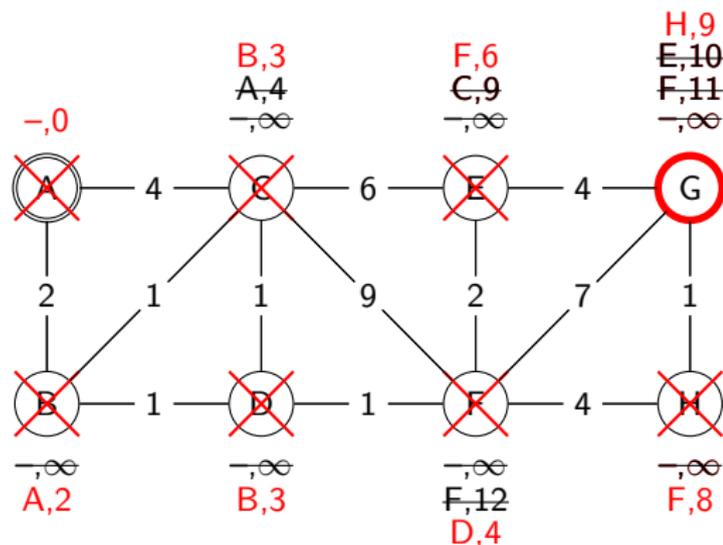
Reihenfolge der Arbeitsknoten: A, B, C, D, F, E, H

Aufgabe 1



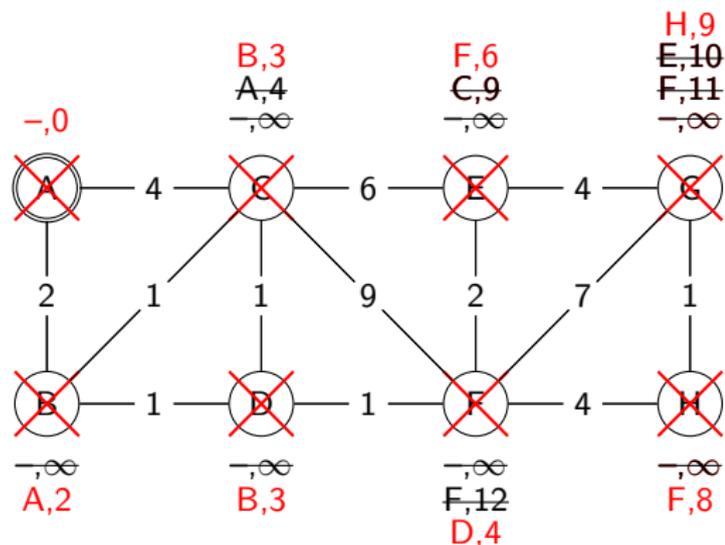
Reihenfolge der Arbeitsknoten: A, B, C, D, F, E, H

Aufgabe 1



Reihenfolge der Arbeitsknoten: A, B, C, D, F, E, H, G

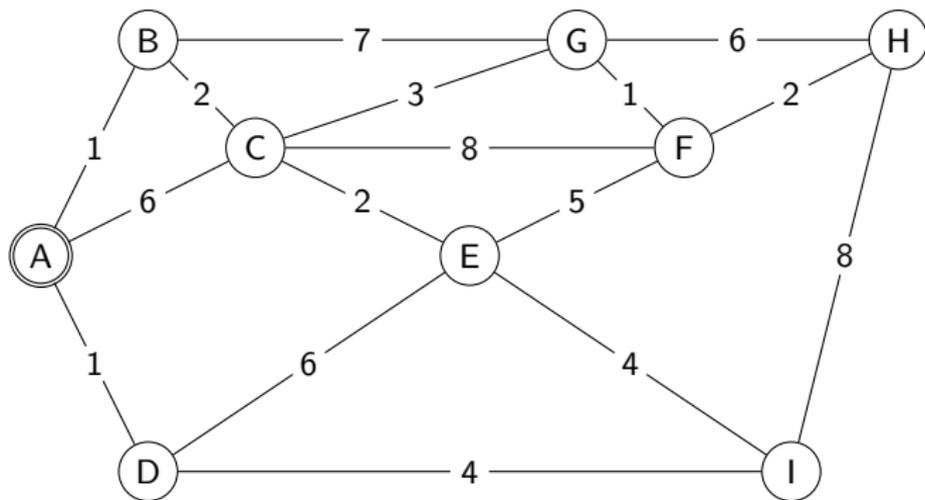
Aufgabe 1



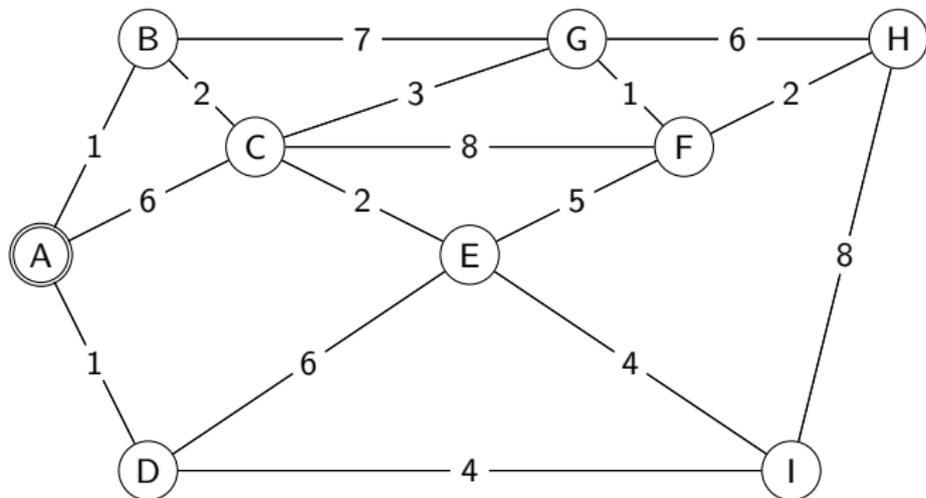
Reihenfolge der Arbeitsknoten: A, B, C, D, F, E, H, G

Aufgabe 2

Bestimme die kürzesten Wege vom Startknoten zu allen übrigen Knoten mit dem Algorithmus von Dijkstra. Protokolliere den Lösungsweg, indem du alle Knoten initialisierst und mit dem Algorithmus aktualisierst.

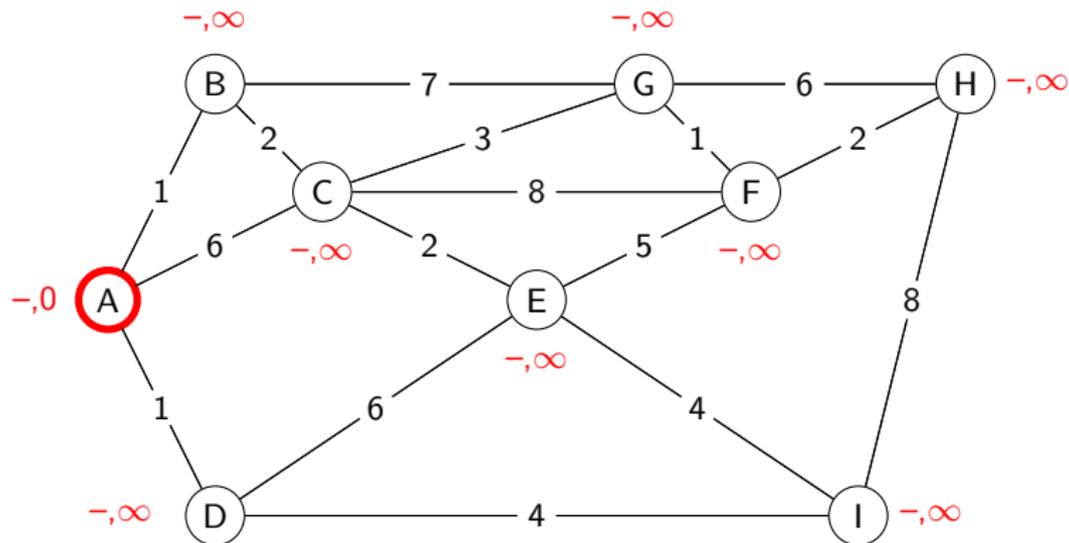


Aufgabe 2



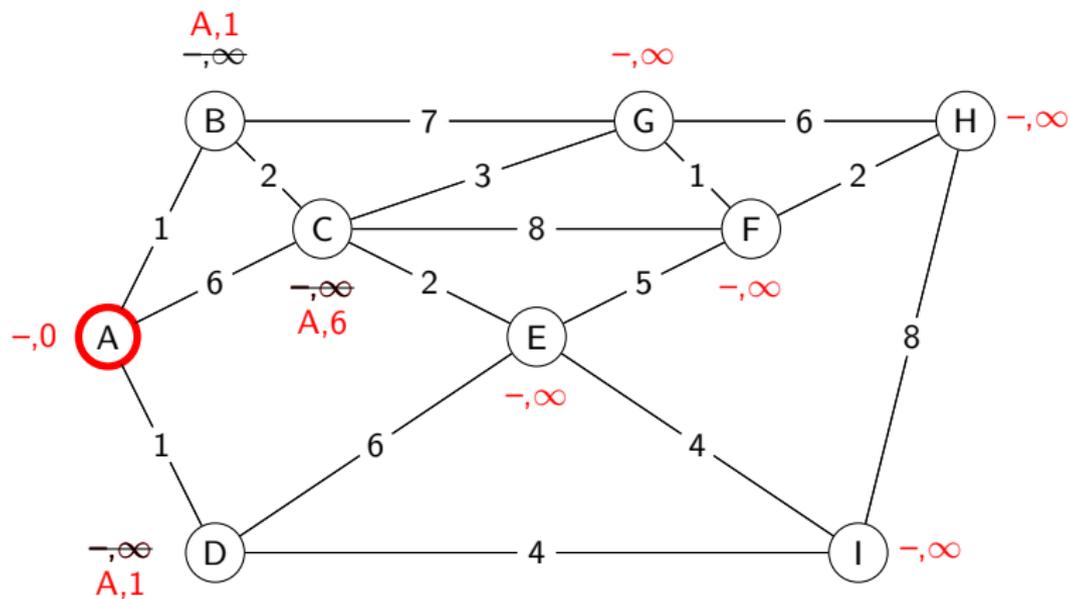
Reihenfolge der Arbeitsknoten:

Aufgabe 2



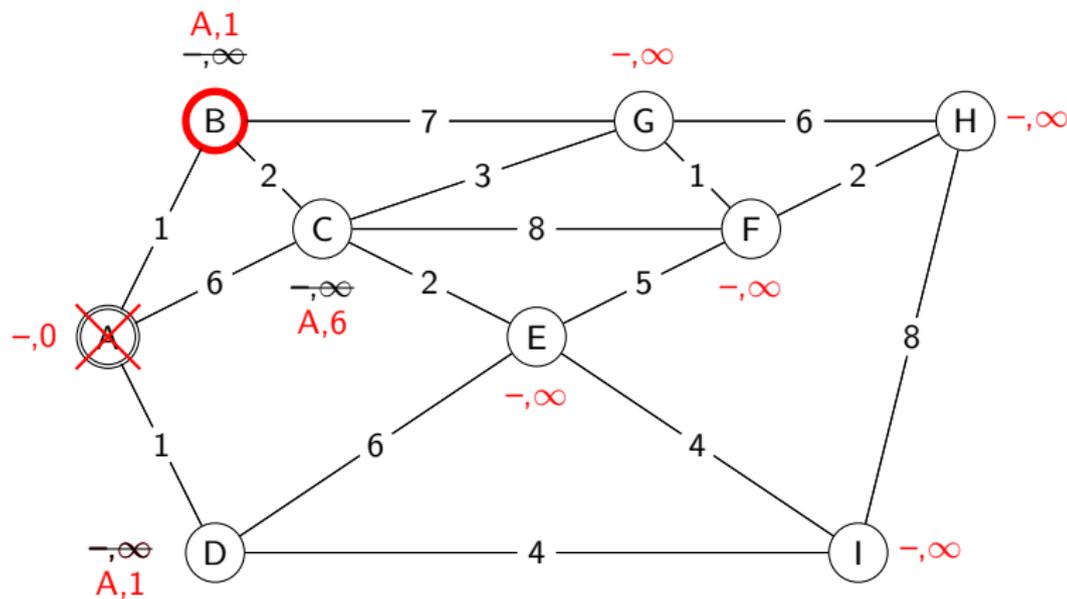
Reihenfolge der Arbeitsknoten: A

Aufgabe 2



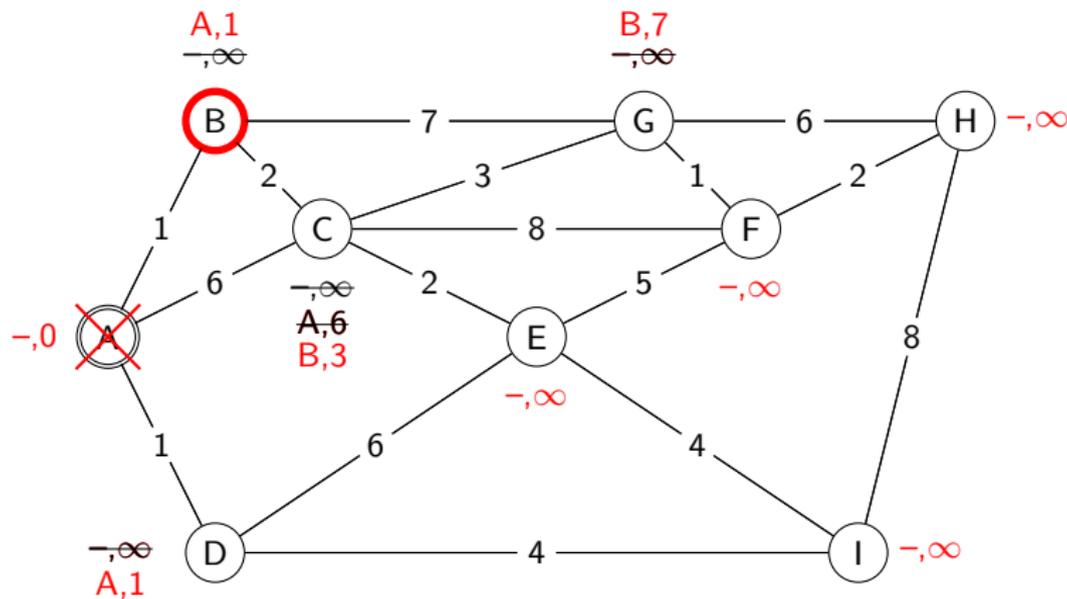
Reihenfolge der Arbeitsknoten: A

Aufgabe 2



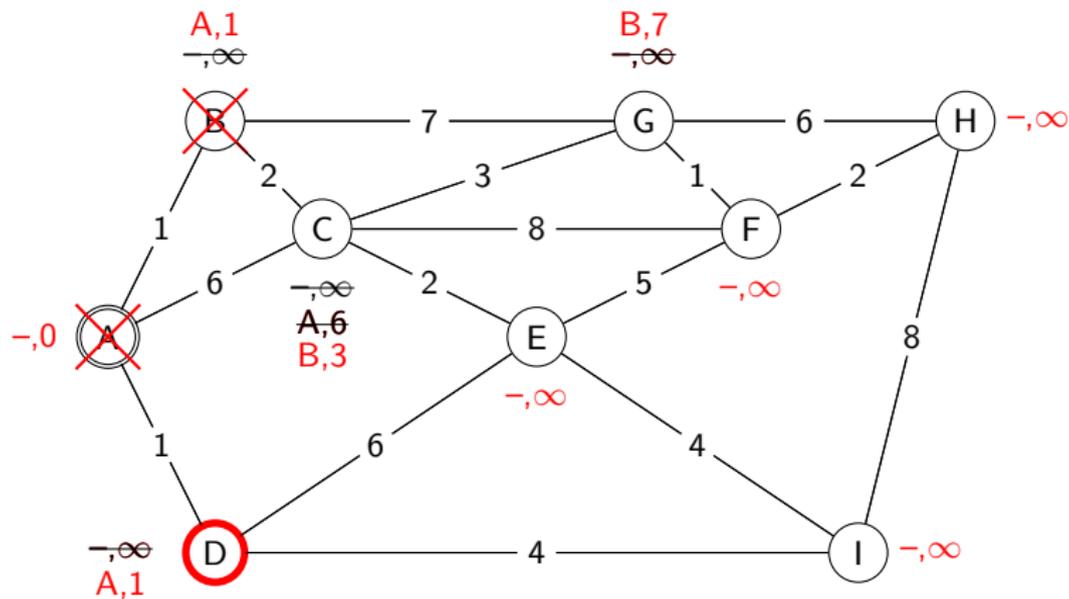
Reihenfolge der Arbeitsknoten: A, B

Aufgabe 2



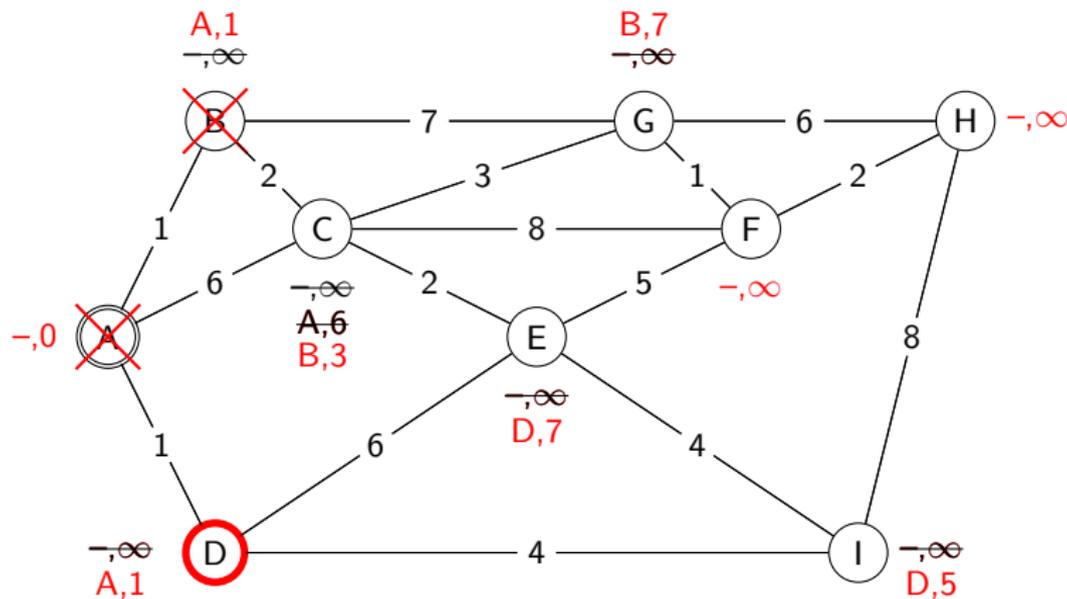
Reihenfolge der Arbeitsknoten: A, B

Aufgabe 2



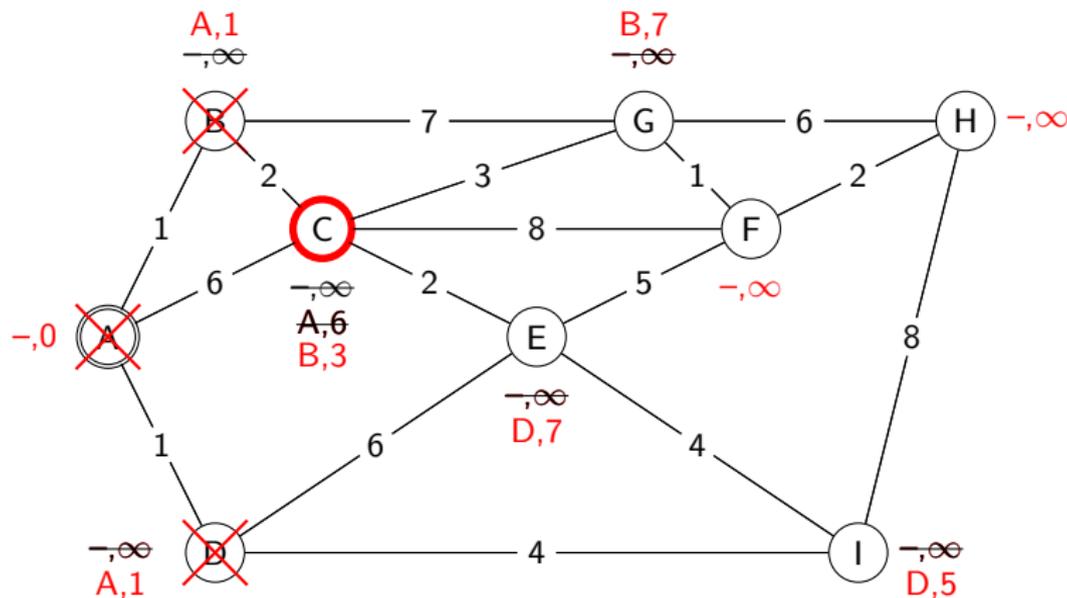
Reihenfolge der Arbeitsknoten: A, B, D

Aufgabe 2



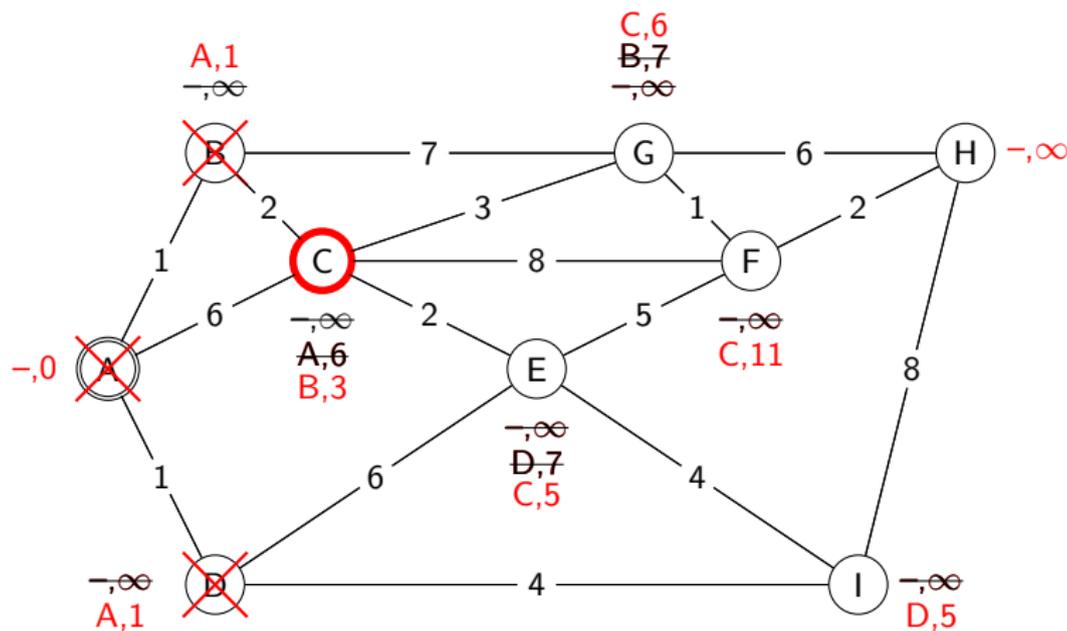
Reihenfolge der Arbeitsknoten: A, B, D

Aufgabe 2



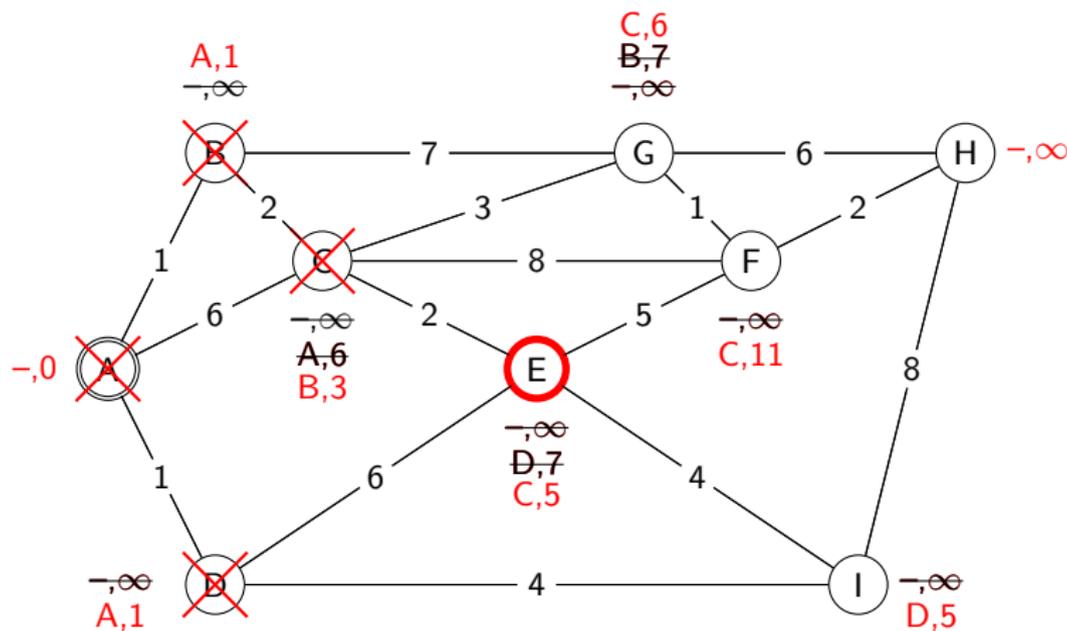
Reihenfolge der Arbeitsknoten: A, B, D, C

Aufgabe 2



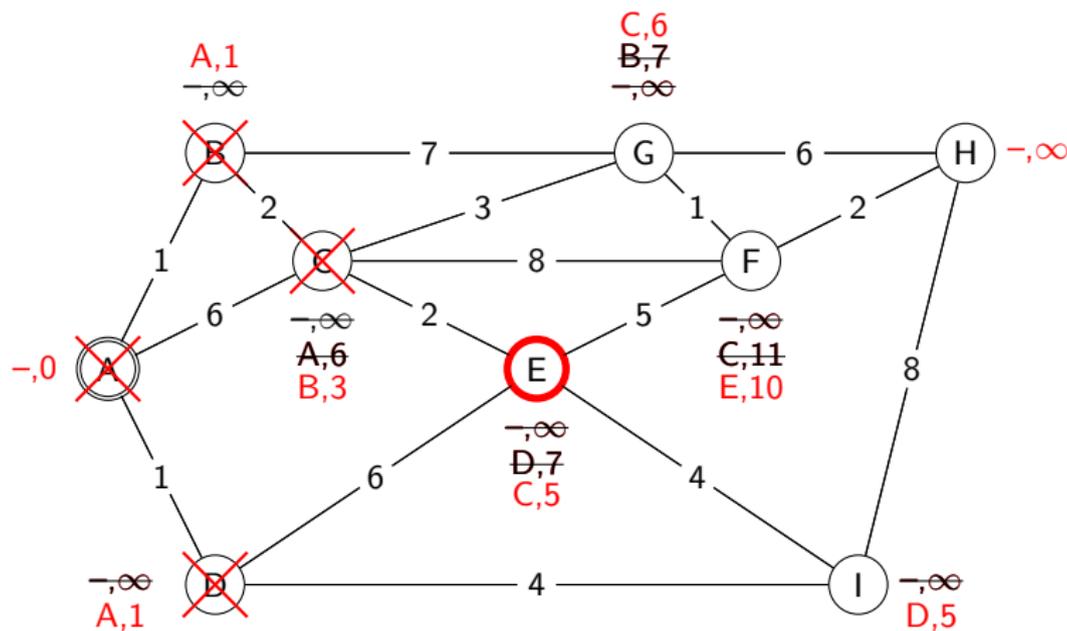
Reihenfolge der Arbeitsknoten: A, B, D, C

Aufgabe 2



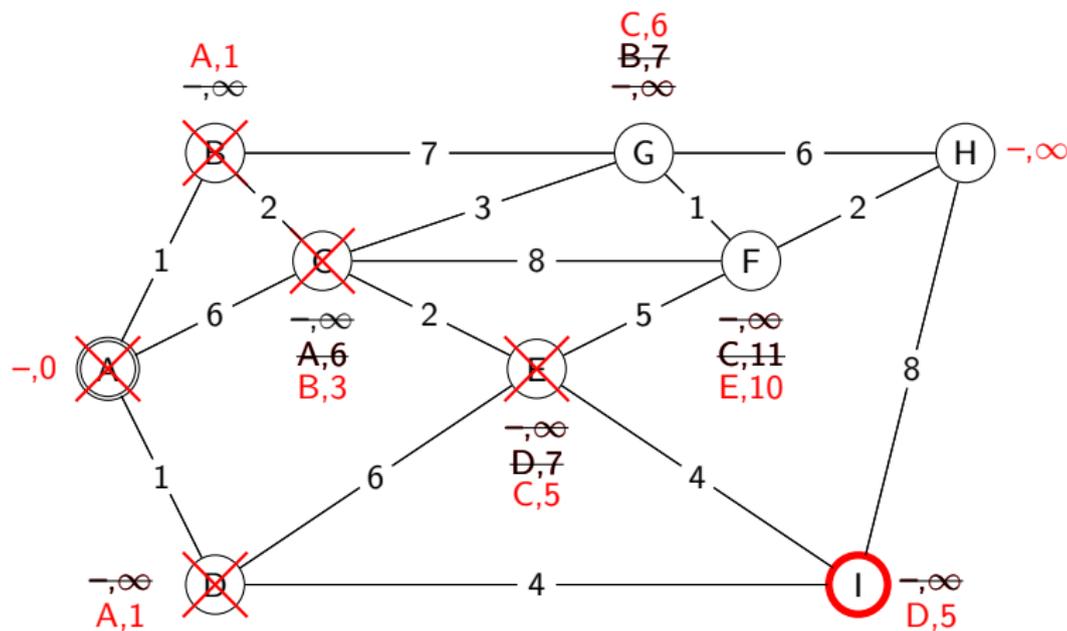
Reihenfolge der Arbeitsknoten: A, B, D, C, E

Aufgabe 2



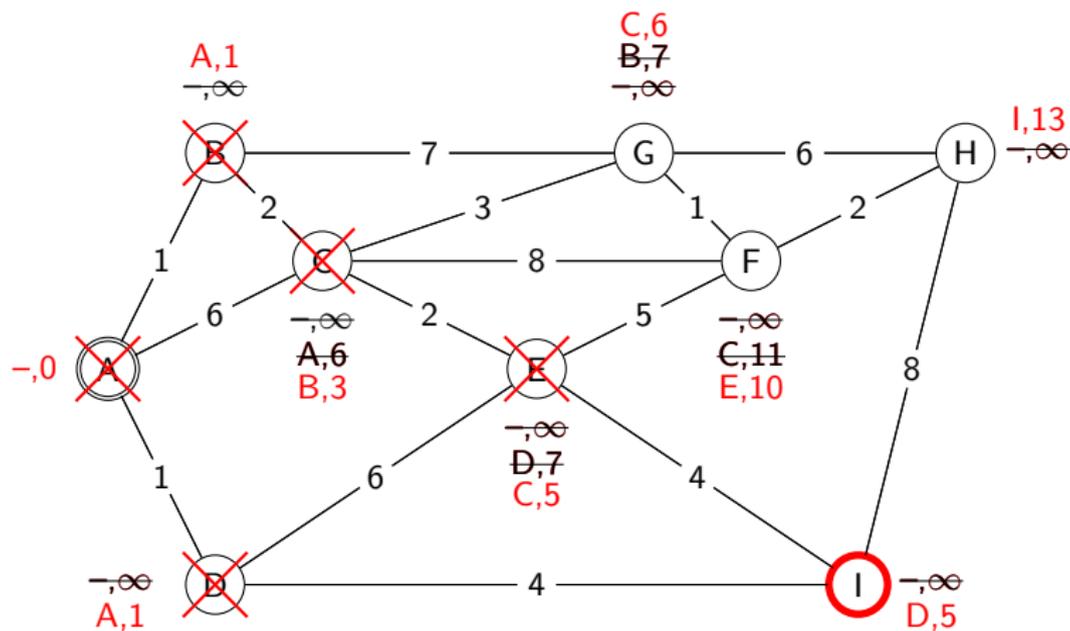
Reihenfolge der Arbeitsknoten: A, B, D, C, E

Aufgabe 2



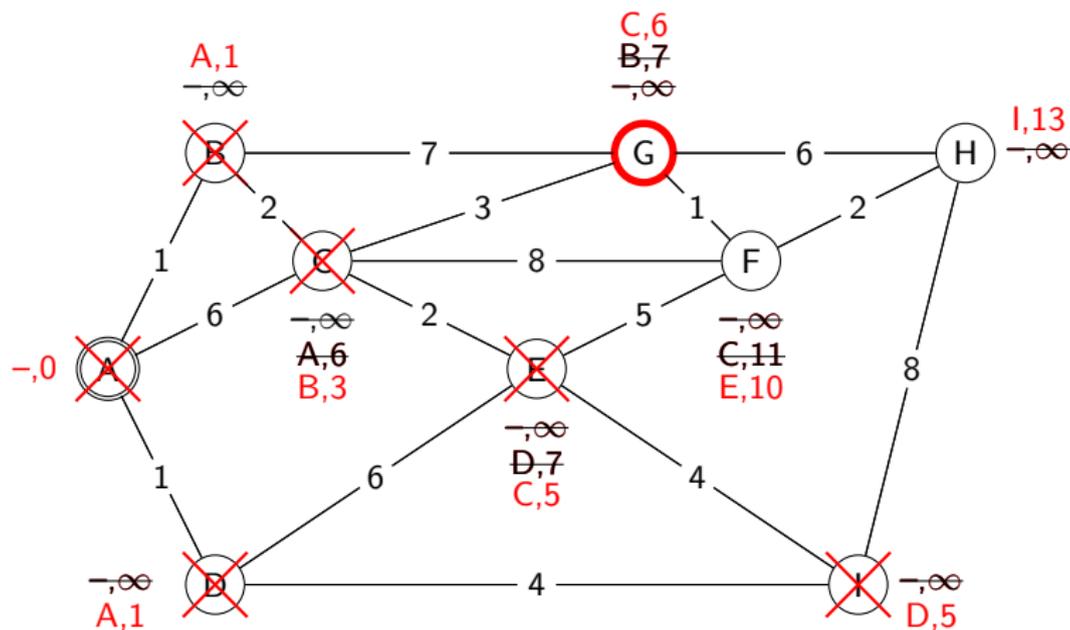
Reihenfolge der Arbeitsknoten: A, B, D, C, E, I

Aufgabe 2



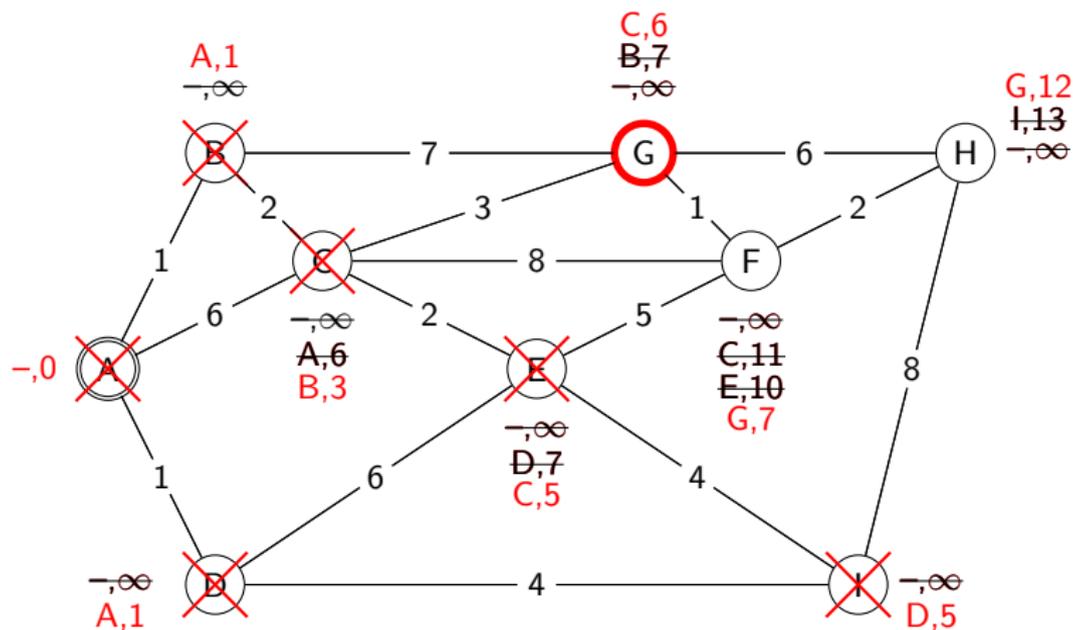
Reihenfolge der Arbeitsknoten: A, B, D, C, E, I

Aufgabe 2



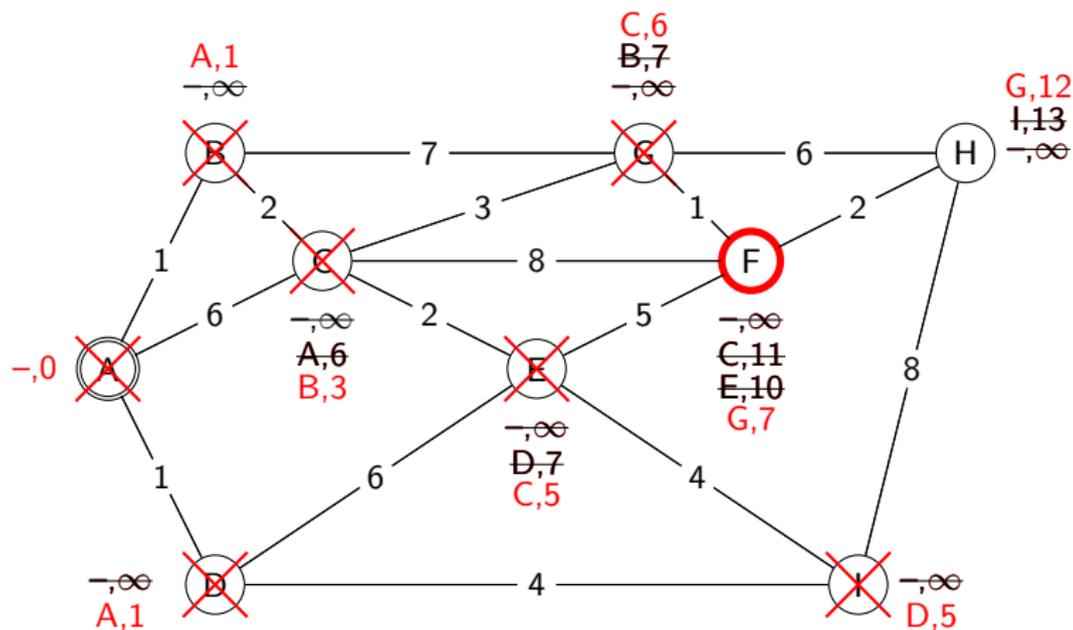
Reihenfolge der Arbeitsknoten: A, B, D, C, E, I, G

Aufgabe 2



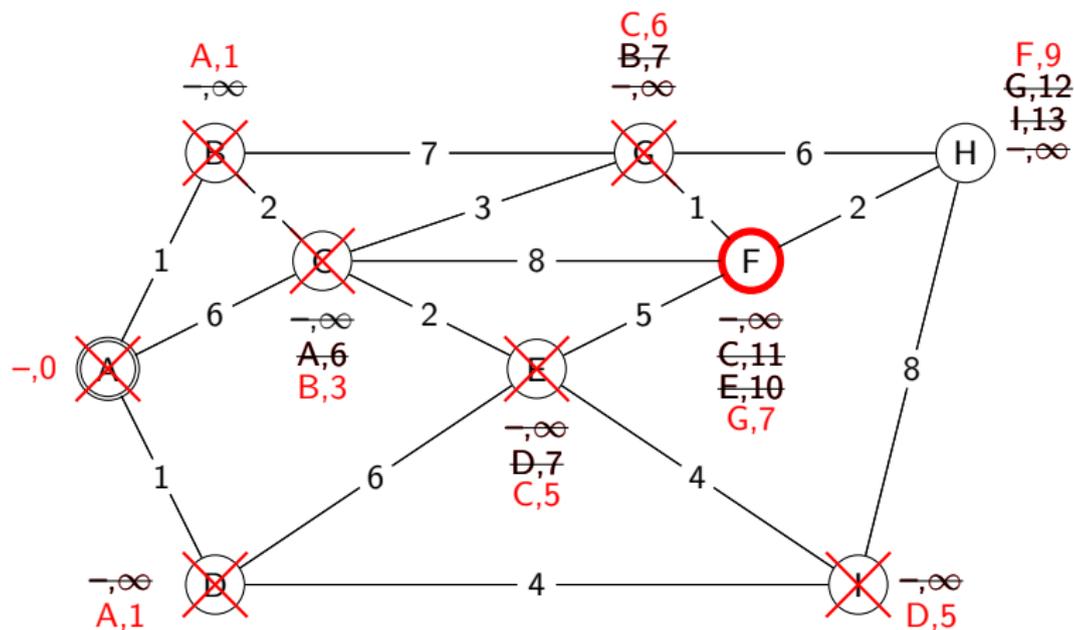
Reihenfolge der Arbeitsknoten: A, B, D, C, E, I, G

Aufgabe 2



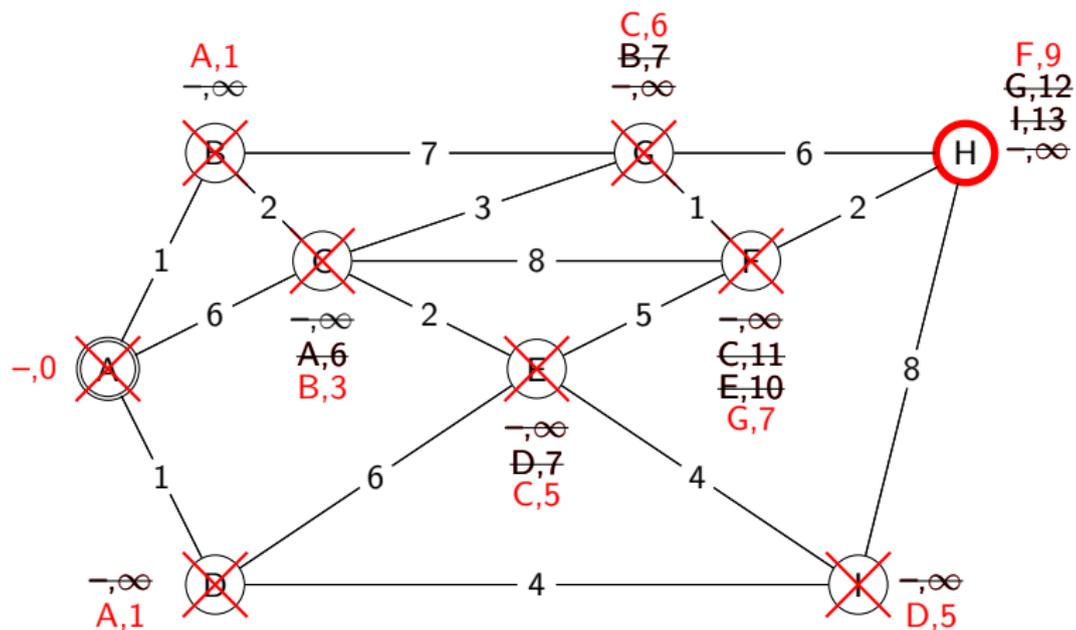
Reihenfolge der Arbeitsknoten: A, B, D, C, E, I, G, F

Aufgabe 2



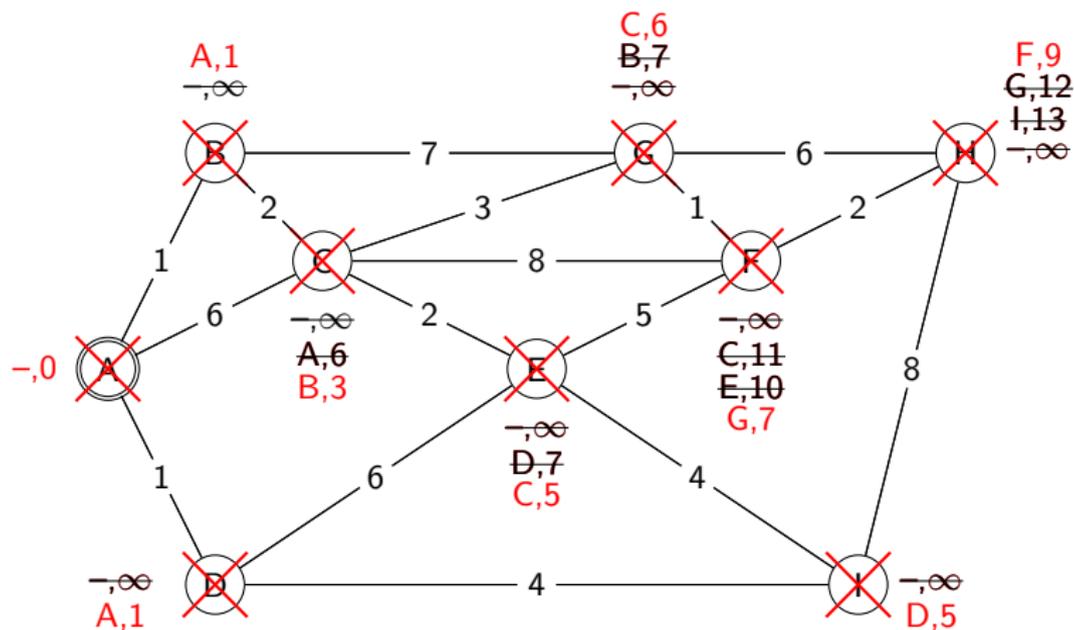
Reihenfolge der Arbeitsknoten: A, B, D, C, E, I, G, F

Aufgabe 2



Reihenfolge der Arbeitsknoten: A, B, D, C, E, I, G, F, H

Aufgabe 2



Reihenfolge der Arbeitsknoten: A, B, D, C, E, I, G, F, H