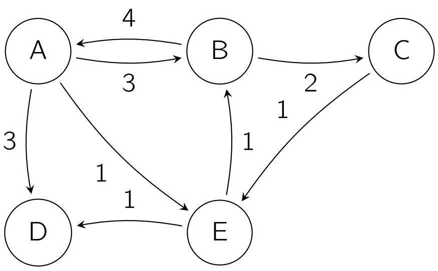
Prof. Dr. Ir. Joost-Pieter Katoen
Christian Dehnert, Friedrich Gretz, Benjamin Kaminski, Thomas Ströder

Tutoraufgabe 1 (Floyd-Warshall):

Betrachten Sie den folgenden Graphen:



Führen Sie den *Algorithmus von Floyd* auf diesem Graphen aus. Geben Sie dazu nach jedem Durchlauf der äußeren Schleife die aktuellen Entfernungen in einer Tabelle an. Die erste Tabelle enthält bereits die Adjazenzmatrix nach Bildung der reflexiven Hülle.Der Eintrag in der Zeile i und Spalte j ist also ∞, falls es keine Kante vom Knoten der Zeile i zu dem Knoten der Spalte j gibt, und sonst das Gewicht dieser Kante. Beachten Sie, dass in der reflexiven Hülle jeder Knoten eine Kante mit Gewicht 0 zu sich selbst hat.

1	A	B	C	D	E
A	0	3	∞	3	1
B	4	0	2	∞	∞
C	∞	∞	0	∞	1
D	∞	∞	∞	0	∞
E	∞	1	∞	1	0

2	A	B	C	D	E
A					
B					
C					
D					
E					

3	A	B	C	D	E
A					
B					
C					
D					
E					

4	A	B	C	D	E
A					
B					
C					
D					
E					

5	A	B	C	D	E
A					
B					
C					
D					
E					

6	A	B	C	D	E
A					
B					
C					
D					
E					

Lösung:

1	A	B	C	D	E
A	0	3	∞	3	1
B	4	0	2	∞	∞
C	∞	∞	0	∞	1
D	∞	∞	∞	0	∞
E	∞	1	∞	1	0

2	A	B	C	D	E
A	0	3	∞	3	1
B	4	0	2	7	5
C	∞	∞	0	∞	1
D	∞	∞	∞	0	∞
E	∞	1	∞	1	0

3	A	B	C	D	E
A	0	3	5	3	1
B	4	0	2	7	5
C	∞	∞	0	∞	1
D	∞	∞	∞	0	∞
E	5	1	3	1	0

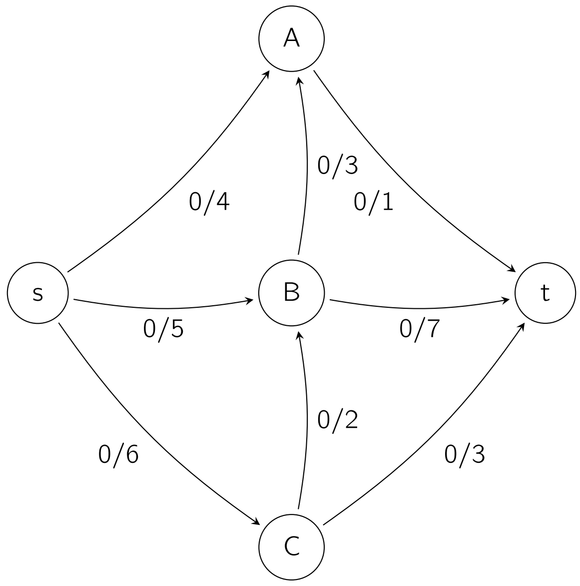
4	A	B	C	D	E
A	0	3	5	3	1
B	4	0	2	7	3
C	∞	∞	0	∞	1
D	∞	∞	∞	0	∞
E	5	1	3	1	0

5	A	B	C	D	E
A	0	3	5	3	1
B	4	0	2	7	3
C	∞	∞	0	∞	1
D	∞	∞	∞	0	∞
E	5	1	3	1	0

6	A	B	C	D	E
A	0	2	4	2	1
B	4	0	2	4	3
C	6	2	0	2	1
D	∞	∞	∞	0	∞
E	5	1	3	1	0

Tutoraufgabe 2 (Ford-Fulkerson):

Betrachten Sie das folgende Flussnetzwerk mit Quelle s und Senke t:

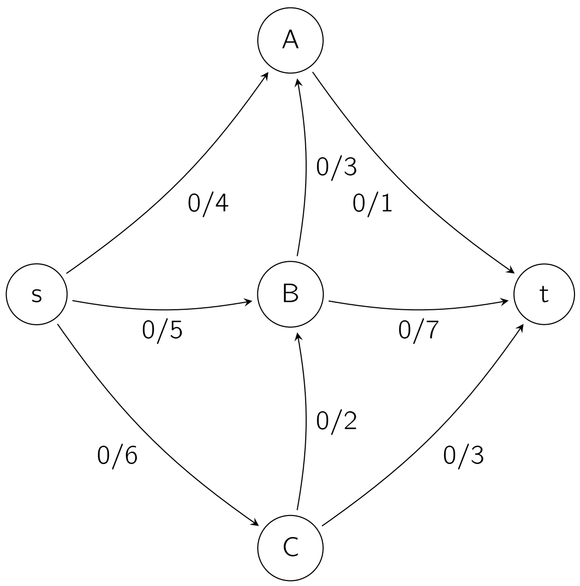


Berechnen Sie den maximalen Fluss in diesem Netzwerk mithilfe der *Ford-Fulkerson Methode*. Geben Sie dazu *jedes Restnetzwerk (auch das initiale)* sowie *nach jeder Augmentierung* den aktuellen Zustand des Flussnetzwerks an. Geben Sie außerdem den *Wert des maximalen Flusses* an. Die vorgegebene Anzahl an Lösungsschritten muss nicht mit der benötigten Anzahl solcher Schritte übereinstimmen.

Lösung:

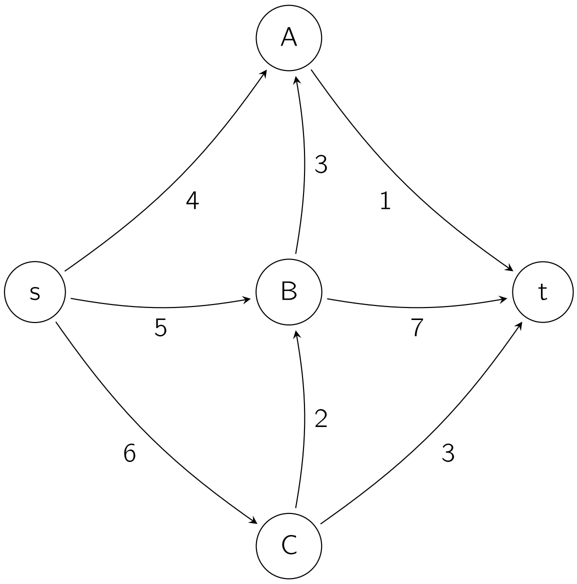
Schritt 0:

Initiales Flussnetzwerk:



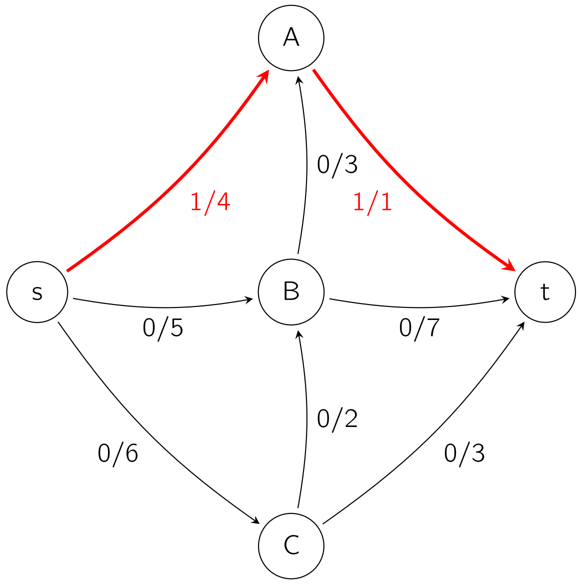
Schritt 1:

Restnetzwerk:



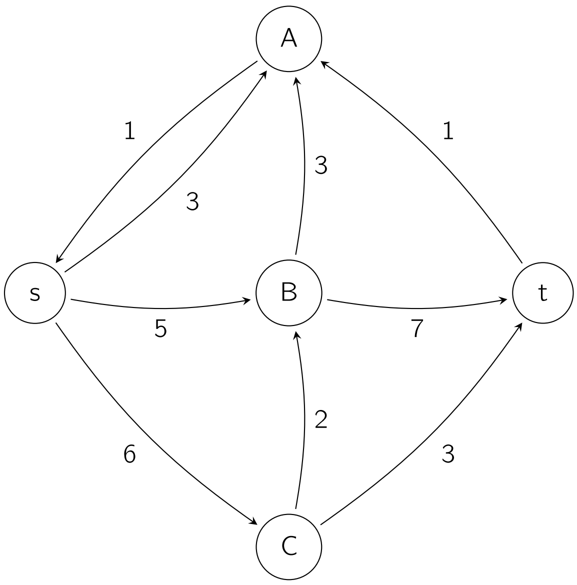
Schritt 2:

Nächstes Flussnetzwerk mit aktuellem Fluss:



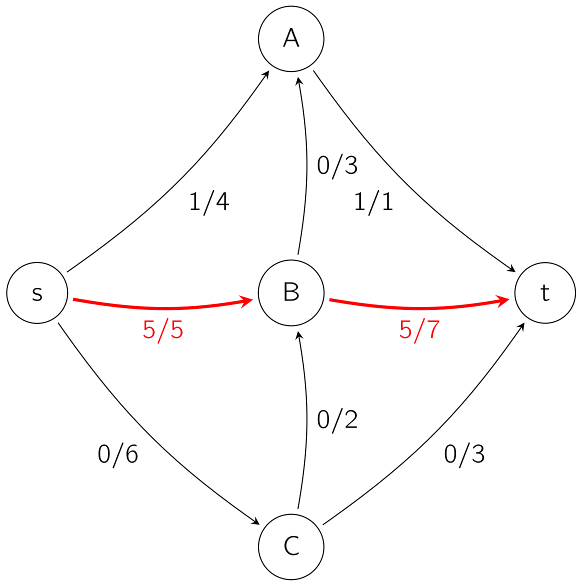
Schritt 3:

Restnetzwerk:



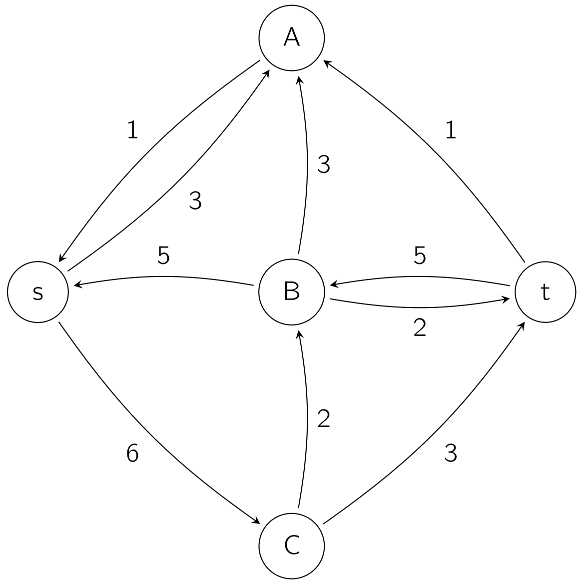
Schritt 4:

Nächstes Flussnetzwerk mit aktuellem Fluss:



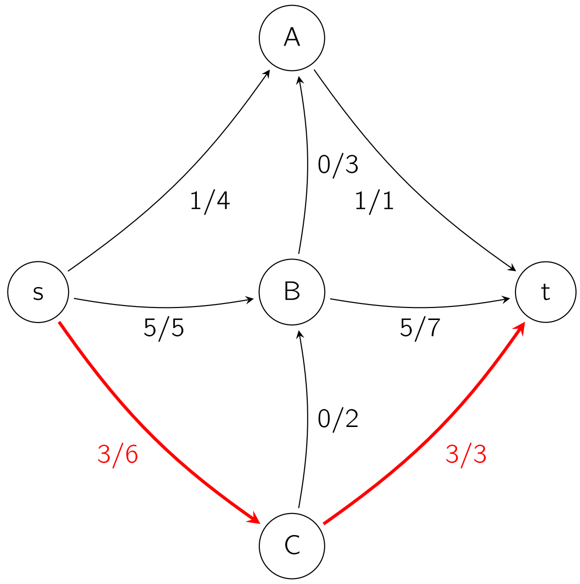
Schritt 5:

Restnetzwerk:



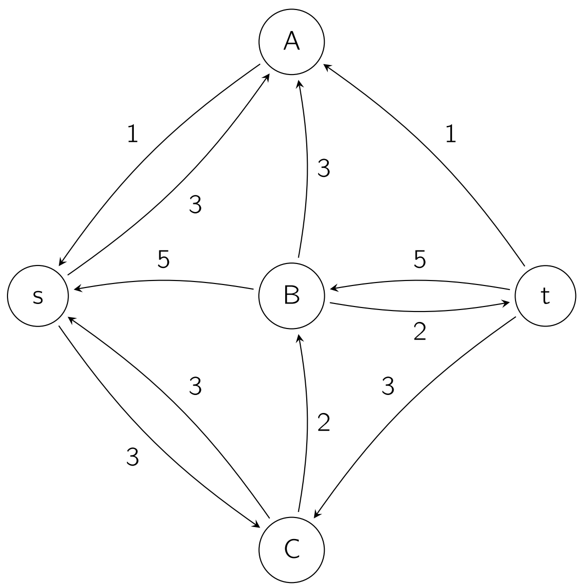
Schritt 6:

Nächstes Flussnetzwerk mit aktuellem Fluss:



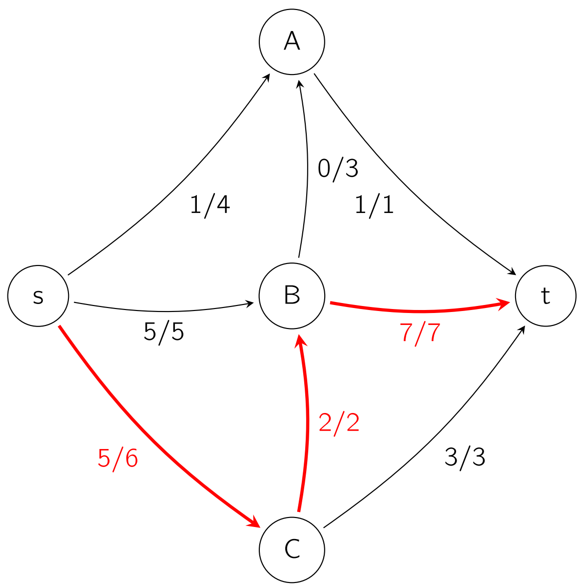
Schritt 7:

Restnetzwerk:



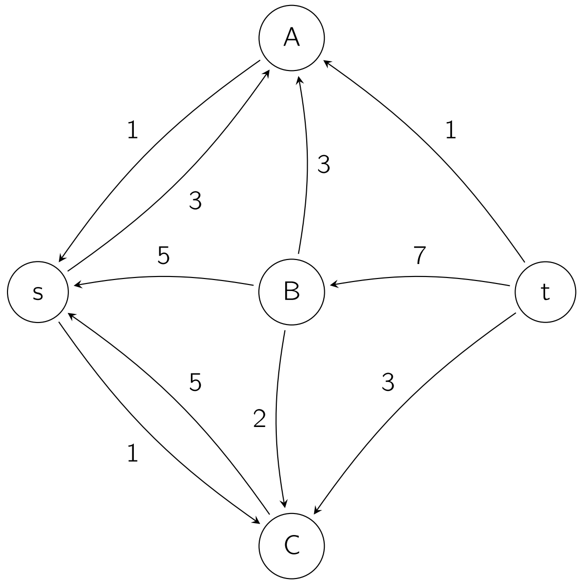
Schritt 8:

Nächstes Flussnetzwerk mit aktuellem Fluss:



Schritt 9:

Restnetzwerk:

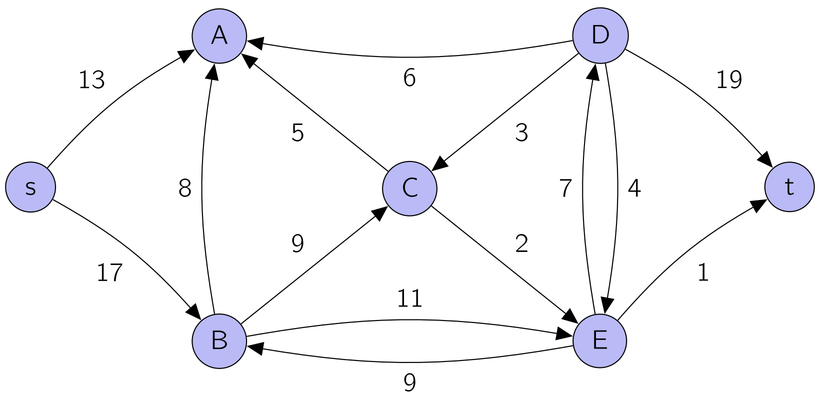


Der maximale Fluss hat den Wert: 11

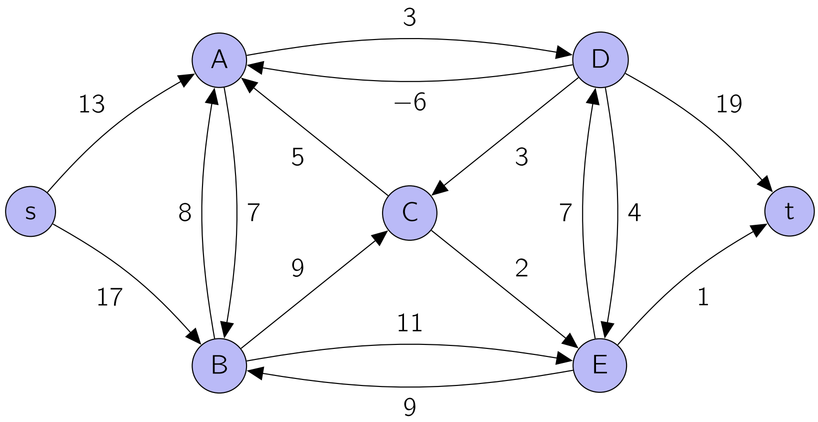
Tutoraufgabe 3 (Flussnetzwerke):

Untersuchen Sie, ob es sich bei den folgenden Graphen um Flussnetzwerke handelt.

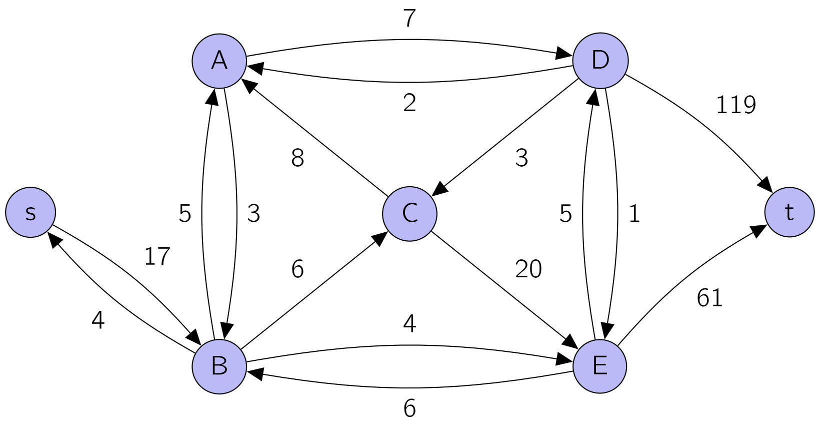
a)



b)



c)



Lösung:

a) Bei diesem Graphen handelt es sich *nicht* um ein Flussnetzwerk, da der Knoten A nicht auf einem Pfad von der Quelle zur Senke liegt.

b) Bei diesem Graphen handelt es sich *nicht* um ein Flussnetzwerk, da die Kante (D, A) eine negative Kapazität, nämlich -6, hat.

c) Bei diesem Graphen handelt es sich um ein Flussnetzwerk, da alle Kapazitäten nichtnegativ sind, lediglich vorhandene Kanten mit einer Kapazität ungleich 0 versehen sind, und alle Knoten auf einem Pfad von der Quelle zur Senke liegen.

Tutoraufgabe 4 (Eigenschaften von Flüssen):

Beweisen Sie die folgenden Aussagen über einen Fluss f und ein Flussnetzwerk $G(V, E, c)$ mit Quelle s und Senke t:

a) Für alle $X, Y \subseteq V$ gilt:

$$f(X, Y) = -f(Y, X)$$

b) Für alle $X, Y, Z \subseteq V$ mit $X \cap Y = \emptyset$ gilt:

$$f(X \cup Y, Z) = f(X, Z) + f(Y, Z)$$

c) Sei (S,T) ein Schnitt über G, so gilt:

$$f(S, T) = |f|$$

d) Sei (S,T) ein Schnitt über G, so gilt:

$$|f| \leq c(S, T)$$

Lösung:

Die folgenden Aussagen waren zu beweisen:

a) Für alle $X, Y \subseteq V$ gilt: $f(X, Y) = -f(Y, X)$

$$\begin{aligned} f(X,Y) &= \sum_{x \in X} \sum_{y \in Y} f(x,y) \\ &= \sum_{x \in X} \sum_{y \in Y} -f(y,x) \\ &= -\sum_{x \in X} \sum_{y \in Y} f(y,x) \\ &= -\sum_{y \in Y} \sum_{x \in X} f(y,x) \\ &= -f(Y,X) \end{aligned}$$

b) Für alle $X, Y, Z \subseteq V$ mit $X \cap Y = \emptyset$ gilt:

$$f(X \cup Y, Z) = f(X, Z) + f(Y, Z)$$

$$\begin{aligned} f(X \cup Y, Z) &= \sum_{x \in X \cup Y} \sum_{y \in Z} f(x,y) && \mid X \cap Y = \emptyset \\ &= \sum_{x \in X} \sum_{y \in Z} f(x,y) + \sum_{x \in Y} \sum_{y \in Z} f(x,y) \\ &= f(X,Z) + f(Y,Z) \end{aligned}$$

c) Sei (S,T) ein Schnitt über G, so gilt: $f(S, T) = |f|$

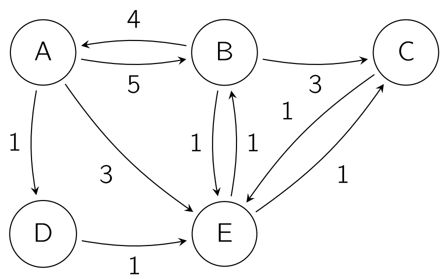
$$\begin{aligned} f(S,T) &= f(S,V) - f(S,V-T) && \mid f(S,V) = f(S,T) + f(S,V-T) \text{ mit } S \cup T = V \\ &= f(S,V) - f(S,S) && \mid f(X,X) = 0 \\ &= f(S,V) && \mid f(X \cup Y, Z) = f(X,Z) + f(Y,Z) \\ &= f(s,V) + f(S - \{s\}, V) && \mid \text{Flusserhaltung} \\ &= f(s,V) = |f| \end{aligned}$$

d) Sei (S,T) ein Schnitt über G, so gilt: $|f| \leq c(S,T)$

$$\begin{aligned}
|f| &= f(S,T) \\
&= \sum_{x \in S} \sum_{y \in T} f(x,y) \\
&\leq \sum_{x \in S} \sum_{y \in T} c(x,y) \\
&= c(S,T)
\end{aligned}$$

Aufgabe 5 (Floyd-Warshall): (5 Punkte)

Betrachten Sie den folgenden Graphen:



Führen Sie den *Algorithmus von Floyd* auf diesem Graphen aus. Geben Sie dazu nach jedem Durchlauf der äußeren Schleife die aktuellen Entfernungen in einer Tabelle an. Die erste Tabelle enthält bereits die Adjazenzmatrix nach Bildung der reflexiven Hülle.Der Eintrag in der Zeile i und Spalte j ist also ∞, falls es keine Kante vom Knoten der Zeile i zu dem Knoten der Spalte j gibt, und sonst das Gewicht dieser Kante. Beachten Sie, dass in der reflexiven Hülle jeder Knoten eine Kante mit Gewicht 0 zu sich selbst hat.

1	A	B	C	D	E
A	0	5	∞	1	3
B	4	0	3	∞	1
C	∞	∞	0	∞	1
D	∞	∞	∞	0	1
E	∞	1	1	∞	0

2	A	B	C	D	E
A					
B					
C					
D					
E					

3	A	B	C	D	E
A					
B					
C					
D					
E					

4	A	B	C	D	E
A					
B					
C					
D					
E					

5	A	B	C	D	E
A					
B					
C					
D					
E					

6	A	B	C	D	E
A					
B					
C					
D					
E					

Lösung:

1	A	B	C	D	E
A	0	5	∞	1	3
B	4	0	3	∞	1
C	∞	∞	0	∞	1
D	∞	∞	∞	0	1
E	∞	1	1	∞	0

2	A	B	C	D	E
A	0	5	∞	1	3
B	4	0	3	5	1
C	∞	∞	0	∞	1
D	∞	∞	∞	0	1
E	∞	1	1	∞	0

3	A	B	C	D	E
A	0	5	8	1	3
B	4	0	3	5	1
C	∞	∞	0	∞	1
D	∞	∞	∞	0	1
E	5	1	1	6	0

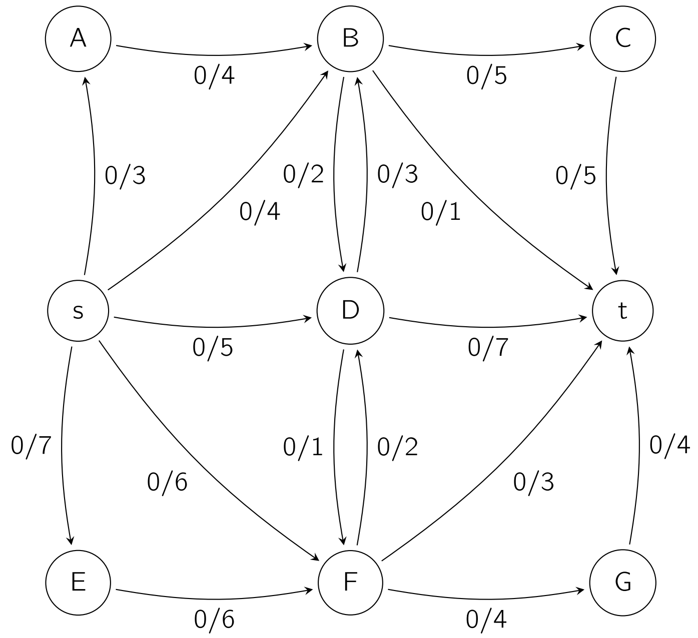
4	A	B	C	D	E
A	0	5	8	1	3
B	4	0	3	5	1
C	∞	∞	0	∞	1
D	∞	∞	∞	0	1
E	5	1	1	6	0

5	A	B	C	D	E
A	0	5	8	1	2
B	4	0	3	5	1
C	∞	∞	0	∞	1
D	∞	∞	∞	0	1
E	5	1	1	6	0

6	A	B	C	D	E
A	0	3	3	1	2
B	4	0	2	5	1
C	6	2	0	7	1
D	6	2	2	0	1
E	5	1	1	6	0

Aufgabe 6 (Ford-Fulkerson): (9 Punkte)

Betrachten Sie das folgende Flussnetzwerk mit Quelle s und Senke t:

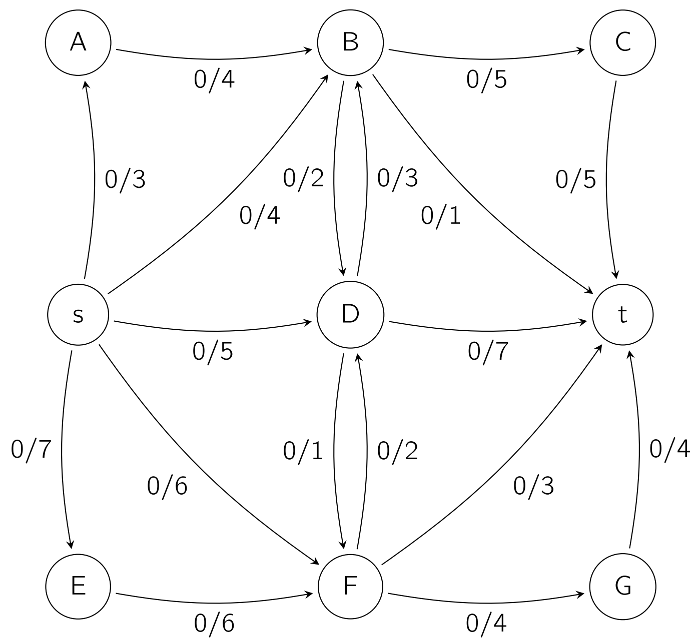


Berechnen Sie den maximalen Fluss in diesem Netzwerk mithilfe der *Ford-Fulkerson Methode*. Geben Sie dazu *jedes Restnetzwerk (auch das initiale)* sowie *nach jeder Augmentierung* den aktuellen Zustand des Flussnetzwerks an. Geben Sie außerdem den *Wert des maximalen Flusses* an. Die vorgegebene Anzahl an Lösungsschritten muss nicht mit der benötigten Anzahl solcher Schritte übereinstimmen.

Lösung:

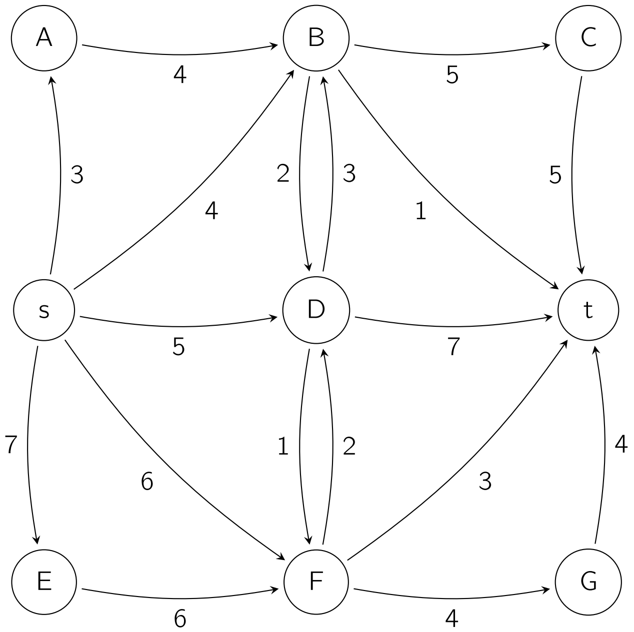
Schritt 0:

Initiales Flussnetzwerk:



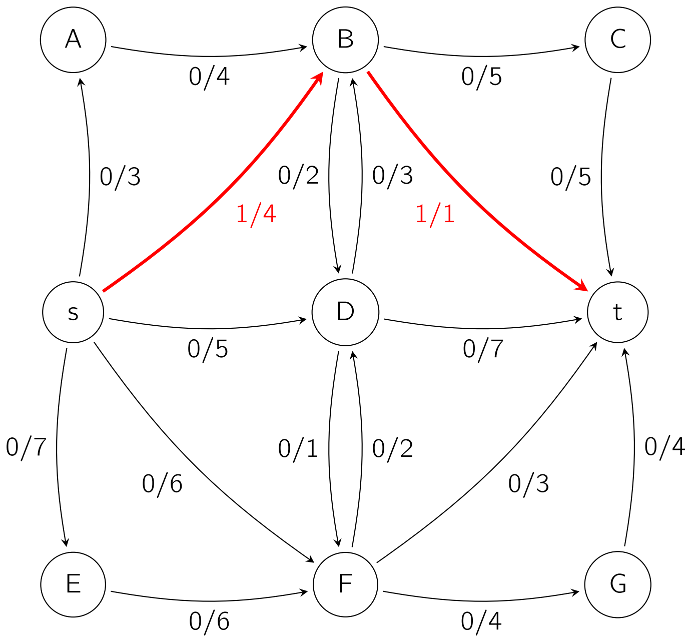
Schritt 1:

Restnetzwerk:



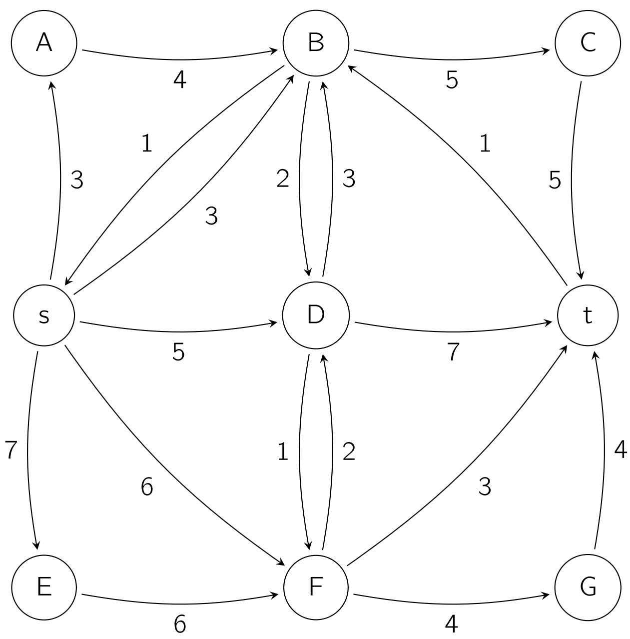
Schritt 2:

Nächstes Flussnetzwerk mit aktuellem Fluss:



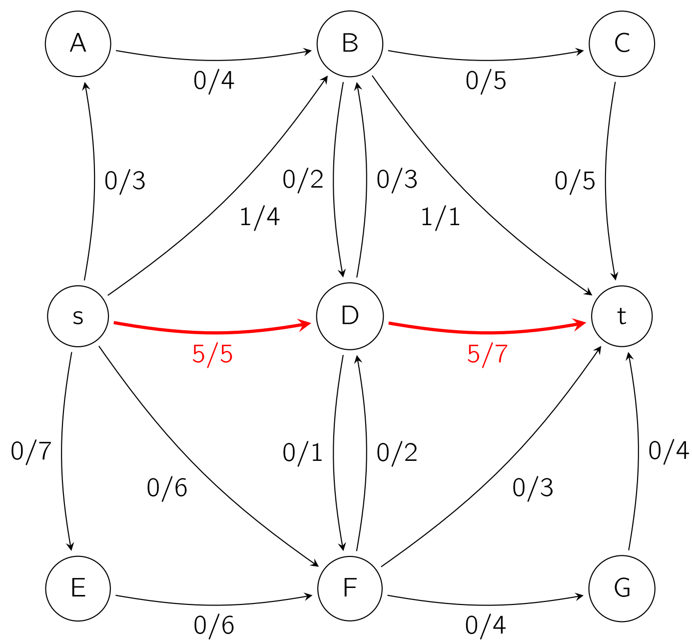
Schritt 3:

Restnetzwerk:



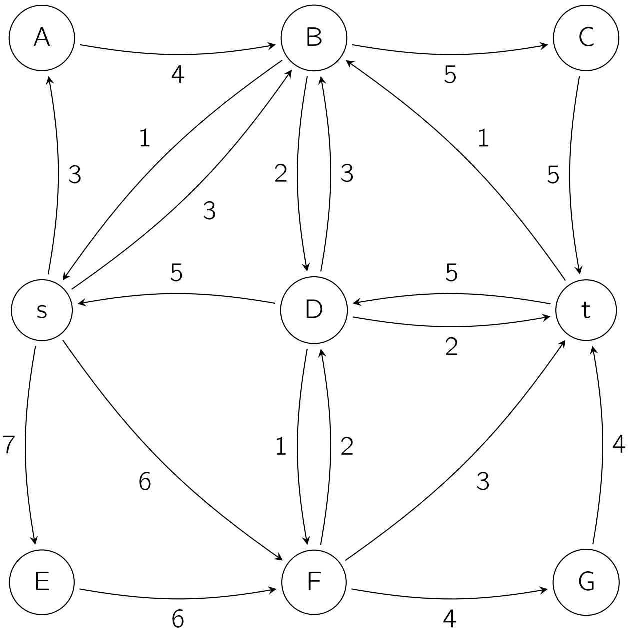
Schritt 4:

Nächstes Flussnetzwerk mit aktuellem Fluss:



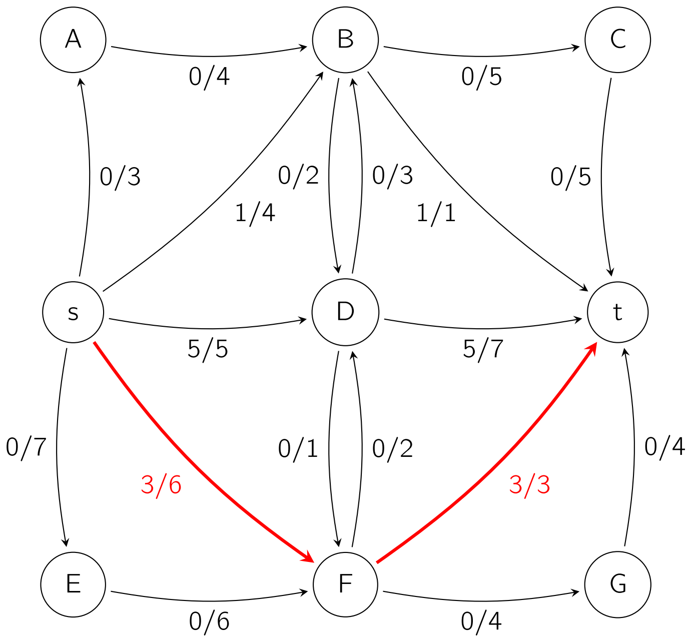
Schritt 5:

Restnetzwerk:



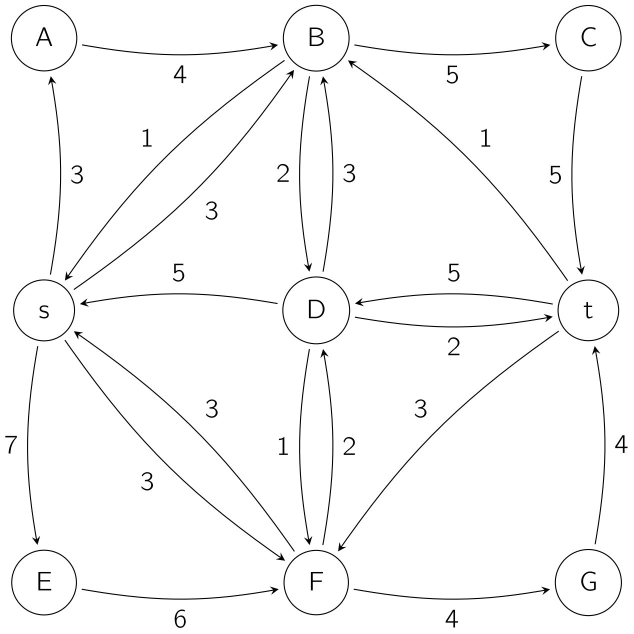
Schritt 6:

Nächstes Flussnetzwerk mit aktuellem Fluss:



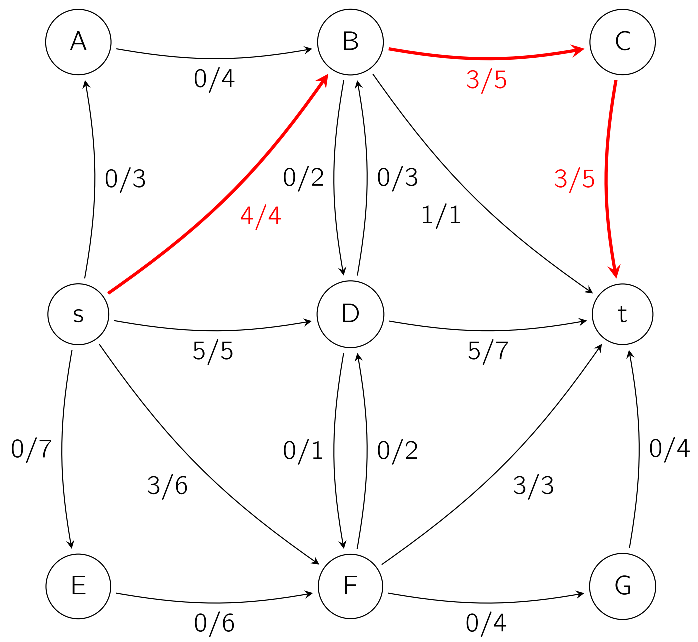
Schritt 7:

Restnetzwerk:



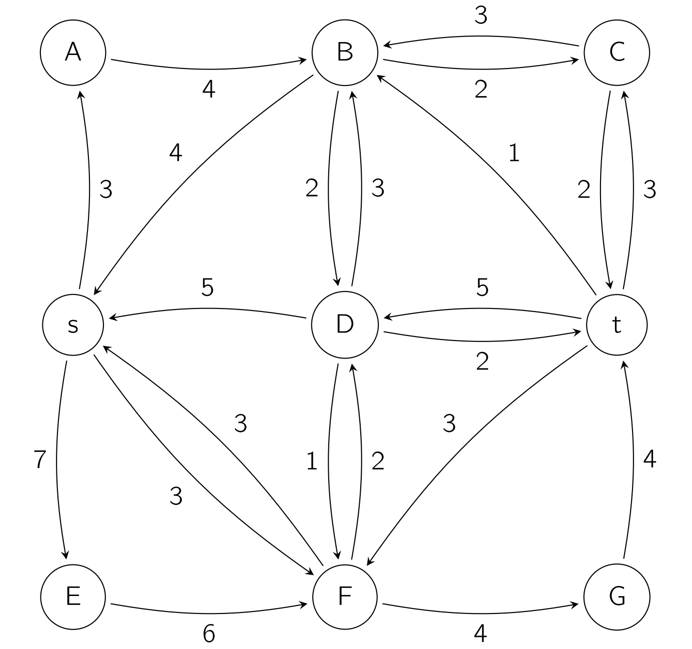
Schritt 8:

Nächstes Flussnetzwerk mit aktuellem Fluss:



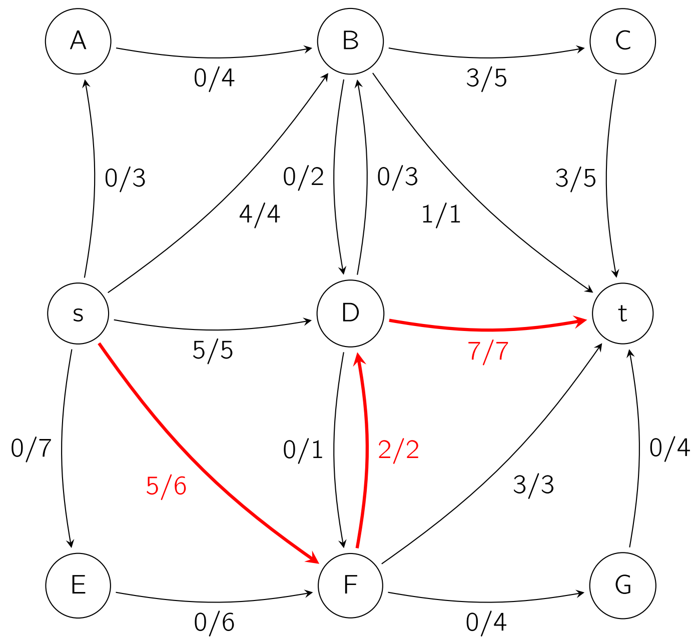
Schritt 9:

Restnetzwerk:



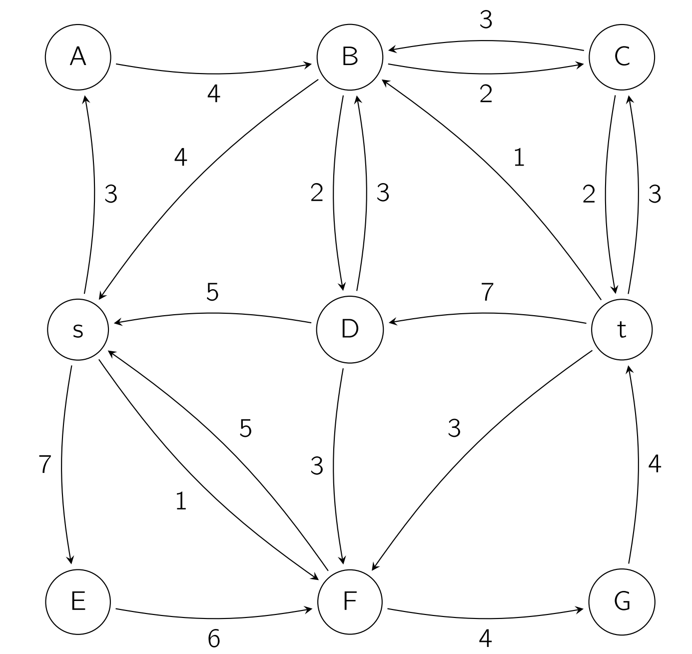
Schritt 10:

Nächstes Flussnetzwerk mit aktuellem Fluss:



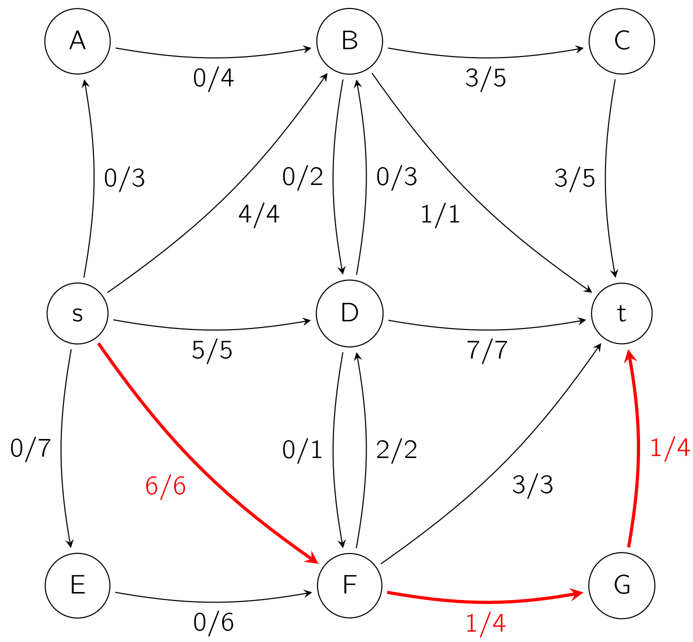
Schritt 11:

Restnetzwerk:



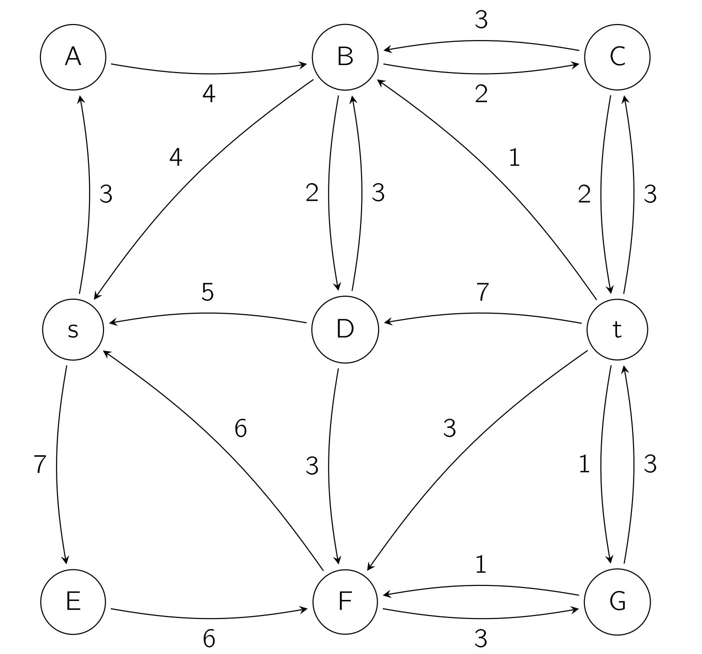
Schritt 12:

Nächstes Flussnetzwerk mit aktuellem Fluss:



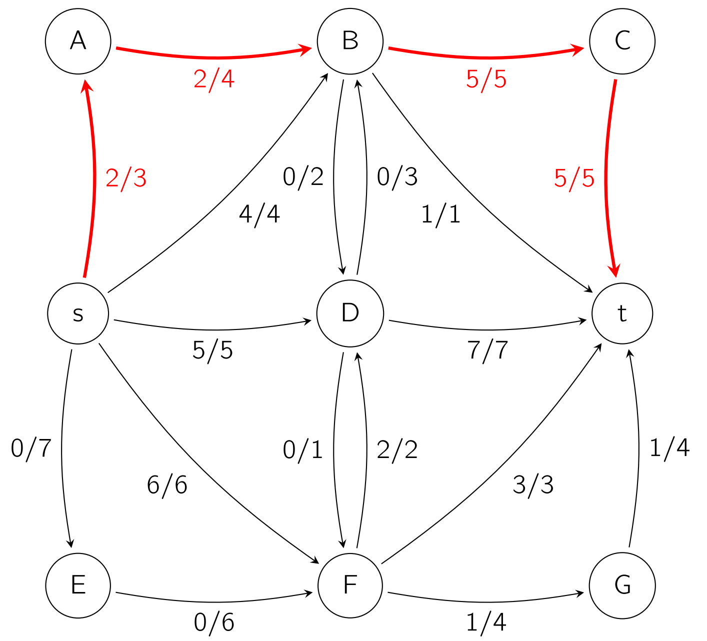
Schritt 13:

Restnetzwerk:



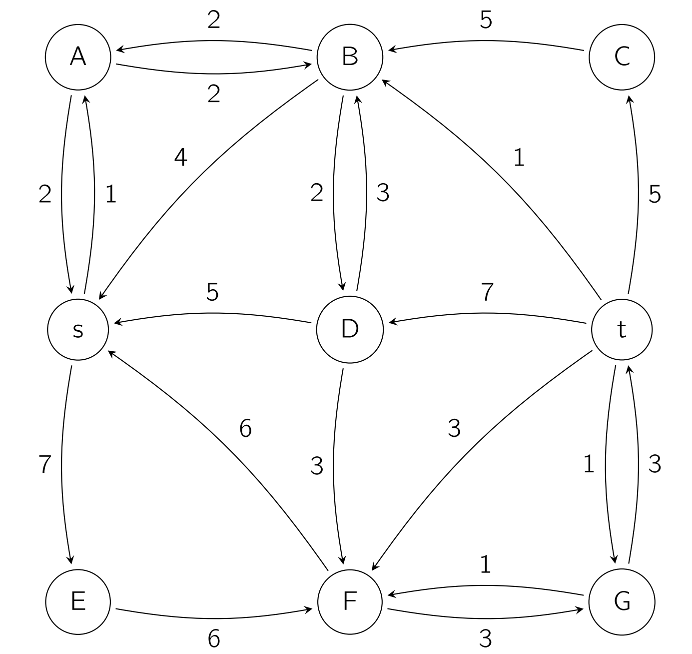
Schritt 14:

Nächstes Flussnetzwerk mit aktuellem Fluss:



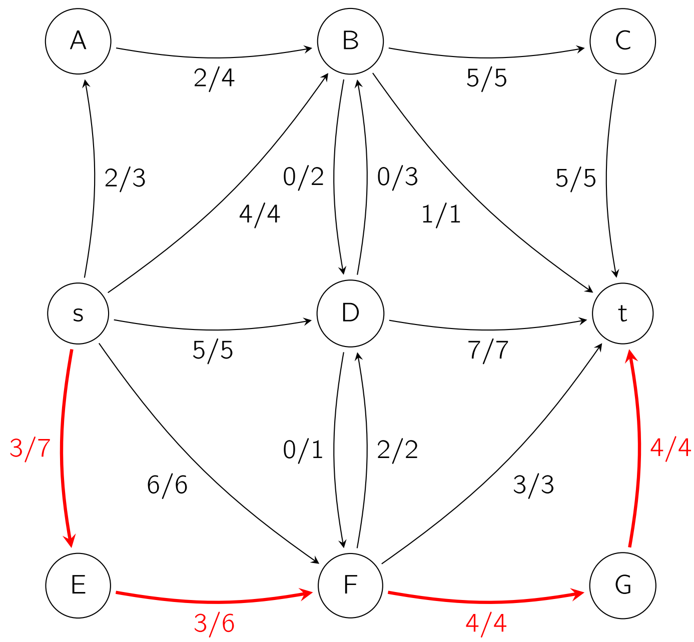
Schritt 15:

Restnetzwerk:



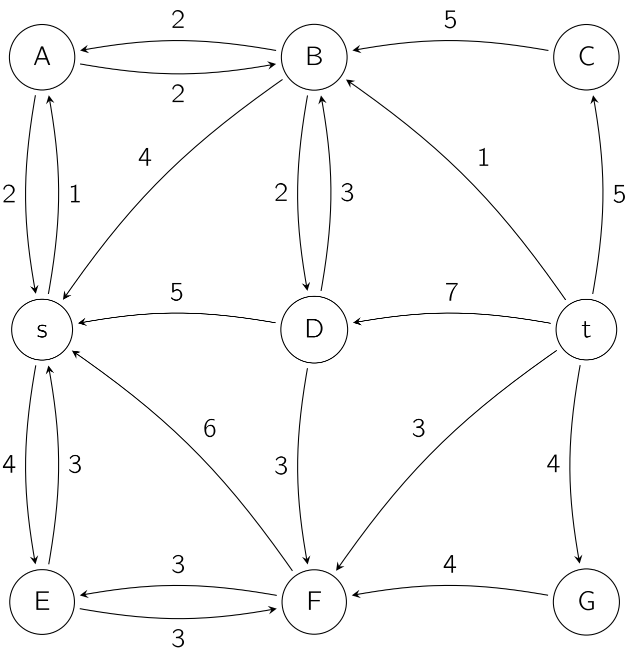
Schritt 16:

Nächstes Flussnetzwerk mit aktuellem Fluss:



Schritt 17:

Restnetzwerk:

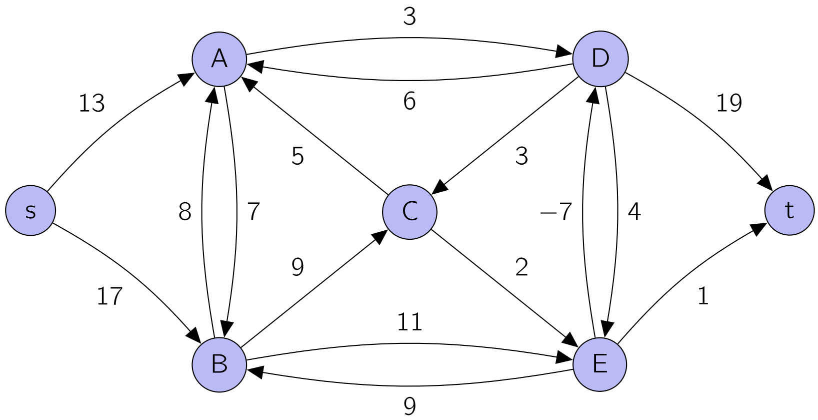


Der maximale Fluss hat den Wert: 20

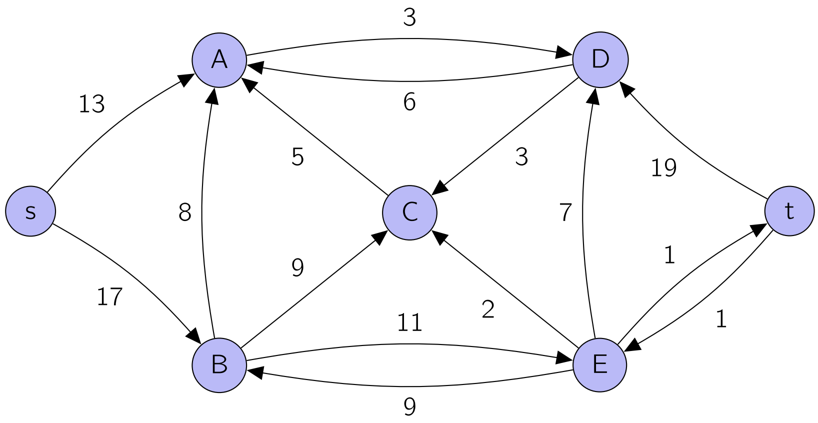
Aufgabe 7 (Flussnetzwerke): (2 + 2 + 2 Punkte)

Untersuchen Sie, ob es sich bei den folgenden Graphen um Flussnetzwerke handelt. Begründen Sie Ihre Antwort.

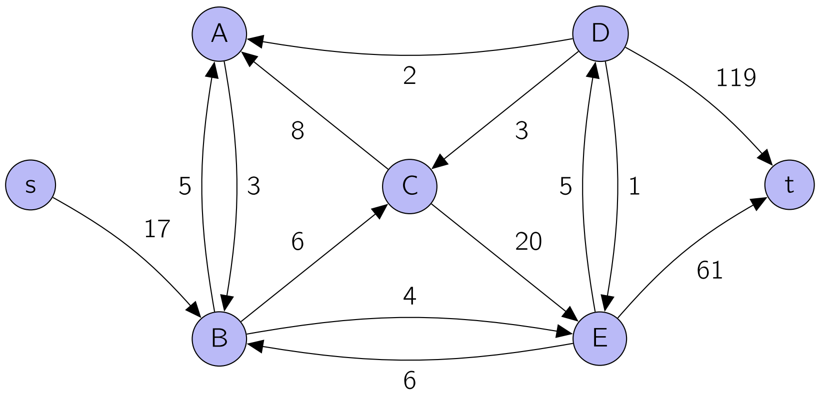
a)



b)



c)



Lösung:

a) Bei diesem Graphen handelt es sich *nicht* um ein Flussnetzwerk, da die Kante $(E,\ D)$ eine negative Kapazität, nämlich -7, hat.

b) Bei diesem Graphen handelt es sich *nicht* um ein Flussnetzwerk, da die Knoten A, C und D nicht auf einem Pfad von der Quelle zur Senke liegen.

c) Bei diesem Graphen handelt es sich um ein Flussnetzwerk, da alle Kapazitäten nichtnegativ sind, lediglich vorhandene Kanten mit einer Kapazität ungleich 0 versehen sind, und alle Knoten auf einem Pfad von der Quelle zur Senke liegen.

Aufgabe 8 (Flussnetzwerke und Flüsse): (10 Punkte)

a) Beweisen oder widerlegen Sie, dass es sich bei dem Graphen

$$\begin{aligned} G &= (V,\ E) \\ V &= \mathbb{N} \cup \{\infty\} \\ E &= \{(x, y) \mid x < y\} \end{aligned}$$

zusammen mit der Kapazitätsfunktion

$$c \colon V \times V \to \mathbb{R},\ (u, v) \mapsto \begin{cases} v, & \text{falls } v = u + 1, \\ 0, & \text{andernfalls.} \end{cases}$$

um ein Flussnetzwerk handelt. Geben Sie für den Fall, dass es sich bei G um ein Flussnetzwerk handelt, die Quelle und die Senke des Flussnetzwerkes an.

b) Betrachten Sie das Flussnetzwerk G aus Teil a). Beweisen oder widerlegen Sie, dass es sich bei der Funktion

$$f : V \times V \to \mathbb{R},\ (u, v) \mapsto \begin{cases} 1, & \text{falls } v = u + 1, \\ -1, & \text{falls } u = v + 1, \\ 0, & \text{andernfalls.} \end{cases}$$

um einen Fluss in G handelt.

Hinweis: Diese Aufgabe wirkt schwerer als sie ist. Zu ihrer Lösung sind größtenteils lediglich Definitionen ineinander einzusetzen. Ziel ist es, Ihnen den Schrecken vor solch kompliziert wirkenden Aufgaben (wie beispielsweise der Treap-Aufgabe in der Präsenzübung) zu nehmen.

Lösung:

a) Bei G handelt es sich um ein Flussnetzwerk mit der Quelle 0 und der Senke ∞.

Wir zeigen dazu zunächst, dass sich jeder Knoten $v \in \mathbb{N} \cup \{\infty\}$ auf einem Pfad von der Quelle zur Senke befindet: Sei dafür $v \in \mathbb{N} \setminus \{0\}$ ein beliebiger Knoten, der nicht die Quelle oder die Senke ist. Dann gilt $0 < v < \infty$ und damit existieren laut definition von E zwei Kanten $(0, v)$ und (v, ∞) und damit auch ein Pfad $(0, v, \infty)$. Somit liegt jeder Knoten auf einem Pfad von der Quelle 0 zur Senke ∞.

Als nächstes Zeigen wir, dass die Kapazitätsfunktion c den Anforderungen an ein Flussnetzwerk entspricht:

- Für alle $(u, v) \in V \times V$ gilt $c(u, v) \geq 0$ und damit gilt insbesondere auch für alle $(u, v) \in E$, dass $c(u, v) \geq 0$.
- Sei $(u, v) \notin E$. Dann gilt:

$$(u, v) \notin E \implies u \geq v \implies v \neq u + 1 \implies c(u, v) = 0$$

b) Bei der Funktion f handelt es sich um einen Fluss. Dazu zeigen wir, dass die drei an einen Fluss gestellten Forderungen gelten:

- **Beschränkung:** Sei $u < v$. Dann folgt, dass $v \geq 1$ ist. Damit gilt $c(u, v) \geq 1$ und damit auch $f(u, v) \leq c(u, v)$, da f maximal den Wert 1 annehmen kann.

  Sei nun $u \geq v$. Dann gilt $f(u, v) \leq 0$ und damit wiederum $f(u, v) \leq c(u, v)$, da c mindestens 0 ist.
- **Asymmetrie:** Sei $v = u + 1$. Dann gilt $f(u, v) = 1$. Dann gilt aber auch $f(v, u) = -1$, denn $v = u + 1$.

  Sei nun $u = v + 1$. Dann gilt $f(u, v) = -1$. Dann gilt aber auch $f(v, u) = 1$, denn $u = v + 1$.

  Gelte nun weder $v = u + 1$ noch $u = v + 1$. Dann gilt entweder $u = v$ oder die Differenz zwischen u und v ist größer 2. In beiden Fällen gilt dann $f(u, v) = 0 = f(v, u)$.
- **Flusserhaltung:** Sei $u \in \mathbb{N} \setminus \{0\}$. Dann gilt:

$$\sum_{v \in V} f(u, v) \;=\; f(u, u-1) + f(u, u+1) + \sum_{v \in V \setminus \{u-1,\, u+1\}} f(u, v) \;=\; -1 + 1 + 0 + 0 + 0 + \cdots \;=\; 0$$

Aufgabe 9 (Sortieren): (10 Minuten, 1 + 1 + 1 + 1 + 2 + 3 = 9 Punkte)

a) Sortieren Sie das folgende Array mithilfe von Selectionsort. Geben Sie dazu das Array nach jeder Swap-Operation an.

7	8	1	4	9	5

Hinweis: Der Pseudocode von Selectionsort wurde auf Übungsblatt 4 in der Tutoraufgabe 1 angegeben.

b) Sortieren Sie das folgende Array mithilfe von Bubblesort. Geben Sie dazu das Array nach jeder Swap-Operation an.

2	6	1	3	7	4

c) Sortieren Sie das folgende Array mithilfe von Insertionsort. Geben Sie dazu das Array nach jeder Iteration der äußeren Schleife an.

6	7	2	4	1	3

d) Sortieren Sie das folgende Array mithilfe von Mergesort. Geben Sie dazu das Array nach jeder Merge-Operation an.

5	1	7	2	4	8	6	3

e) Sortieren Sie das folgende Array mithilfe von Quicksort. Geben Sie dazu das Array nach jeder Partition-Operation an und markieren Sie das jeweils verwendete Pivot-Element.

2	8	3	7	1	4	9	5

f) Sortieren Sie das folgende Array mithilfe von Heapsort. Geben Sie dazu das Array nach jeder Swap-Operation an.

3	7	1	4	6	5

Lösung:

a)

7	8	1	4	9	5
1	8	7	4	9	5
1	4	7	8	9	5
1	4	5	8	9	7
1	4	5	7	9	8
1	4	5	7	8	9

b)

2	6	1	3	7	4
2	1	6	3	7	4
2	1	3	6	7	4
2	1	3	6	4	7
1	2	3	6	4	7
1	2	3	4	6	7

6	7	2	4	1	3
6	7	2	4	1	3
2	6	7	4	1	3
2	4	6	7	1	3
1	2	4	6	7	3
1	2	3	4	6	7

c)

5	1	7	2	4	8	6	3
1	5	7	2	4	8	6	3
1	5	2	7	4	8	6	3
1	2	5	7	4	8	6	3
1	2	5	7	4	8	6	3
1	2	5	7	4	8	3	6
1	2	5	7	3	4	6	8
1	2	3	4	5	6	7	8

d)

2	8	3	7	1	4	9	5
2	4	3	1	5	8	9	7
1	4	3	2	5	8	9	7
1	2	3	4	5	8	9	7
1	2	3	4	5	8	9	7
1	2	3	4	5	7	9	8
1	2	3	4	5	7	8	9

e)

3	7	1	4	6	5
3	7	5	4	6	1
7	3	5	4	6	1
7	6	5	4	3	1
1	6	5	4	3	7
6	1	5	4	3	7
6	4	5	1	3	7
3	4	5	1	6	7
5	4	3	1	6	7
1	4	3	5	6	7
4	1	3	5	6	7
3	1	4	5	6	7
1	3	4	5	6	7

f)