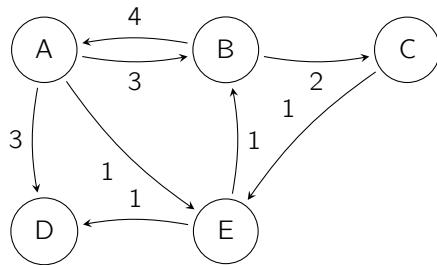


Allgemeine Hinweise:

- Die **Hausaufgaben** sollen in Gruppen von je **2-3 Studierenden** aus der **gleichen Kleingruppenübung (Tutorium)** bearbeitet werden. **Namen und Matrikelnummern** der Studierenden sind auf jedes Blatt der Abgabe zu schreiben. **Heften bzw. tackern Sie die Blätter!**
  - Die **Nummer der Übungsgruppe** muss **links oben** auf das **erste Blatt** der Abgabe geschrieben werden. Notieren Sie die Gruppennummer gut sichtbar, damit wir besser sortieren können.
  - Die Lösungen müssen **bis Mittwoch, den 08.07.2015 um 12:15 Uhr** in den entsprechenden Übungskästen eingeworfen werden. Sie finden die Kästen am Eingang Halifaxstr. des Informatikzentrums (Ahornstr. 55). Alternativ können Sie die Lösungen auch vor der Abgabefrist direkt bei Ihrer Tutorin/Ihrem Tutor abgeben.
  - In Aufgaben, bei denen Sie Algorithmen implementieren sollen, dürfen Sie Ihre Lösung als Pseudo-Code abgeben. Abgaben in verbreiteten imperativen Sprachen wie Java oder C++ sind ebenfalls erlaubt.
- 

### 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 Entfernungswerte 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  $\infty$ , 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	$\infty$	3	1
B	4	0	2	$\infty$	$\infty$
C	$\infty$	$\infty$	0	$\infty$	1
D	$\infty$	$\infty$	$\infty$	0	$\infty$
E	$\infty$	1	$\infty$	1	0

(2)	A	B	C	D	E
A					
B					
C					
D					
E					

(3)	A	B	C	D	E
A					
B					
C					
D					
E					

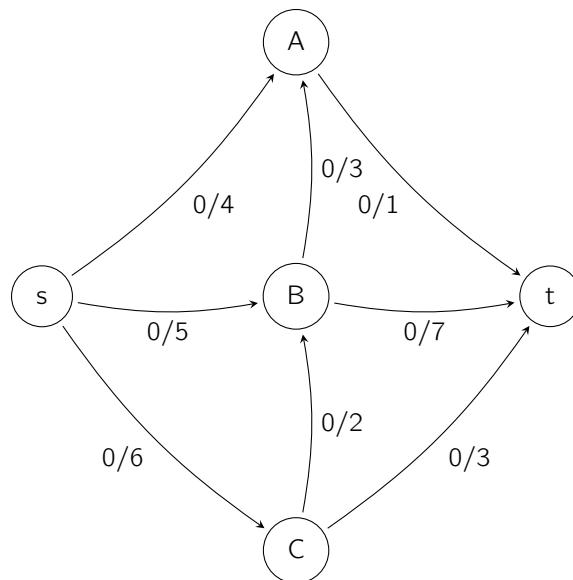
(4)	A	B	C	D	E
A					
B					
C					
D					
E					

(5)	A	B	C	D	E
A					
B					
C					
D					
E					

(6)	A	B	C	D	E
A					
B					
C					
D					
E					

### Tutoraufgabe 2 (Ford-Fulkerson):

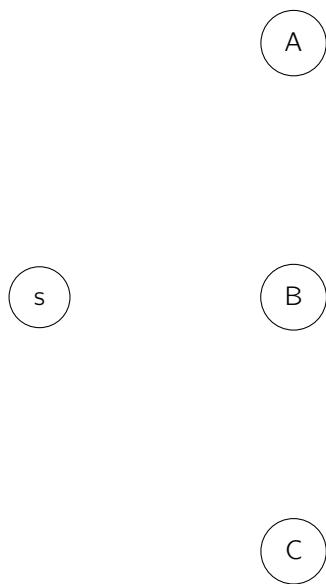
Betrachten Sie das folgende Flussnetzwerk mit Quelle s und Senke t:



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.

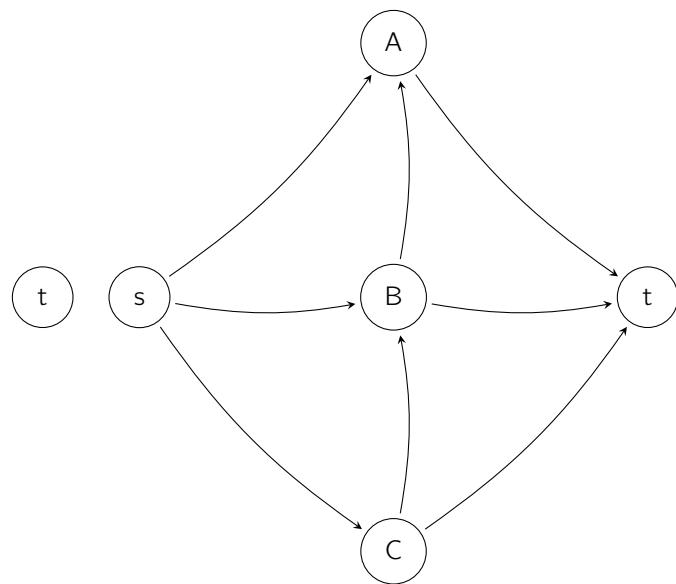
Schritt 1:

Restnetzwerk:



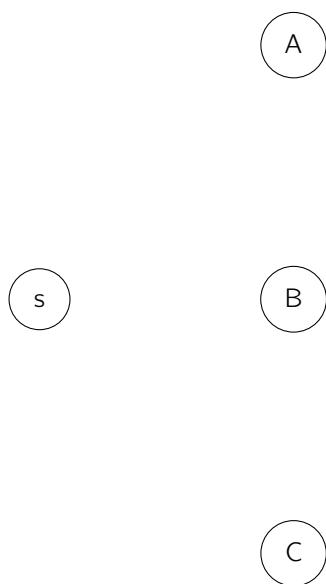
Schritt 2:

Nächstes Flussnetzwerk mit aktuellem Fluss:



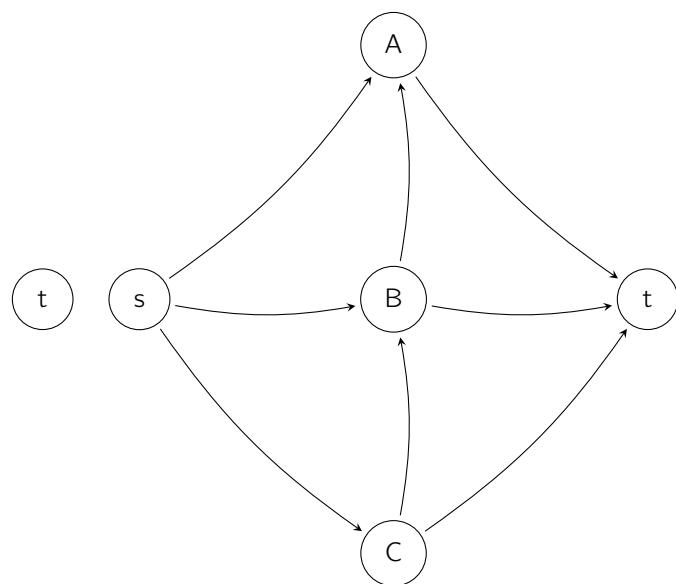
Schritt 3:

Restnetzwerk:



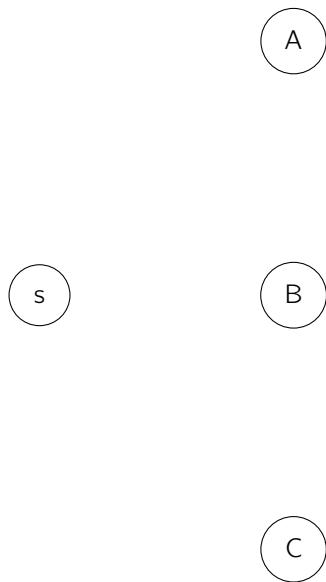
Schritt 4:

Nächstes Flussnetzwerk mit aktuellem Fluss:



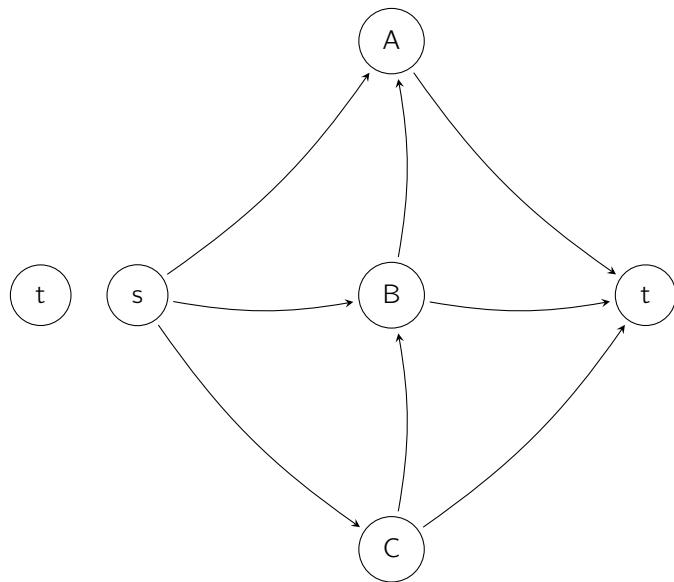
Schritt 5:

Restnetzwerk:



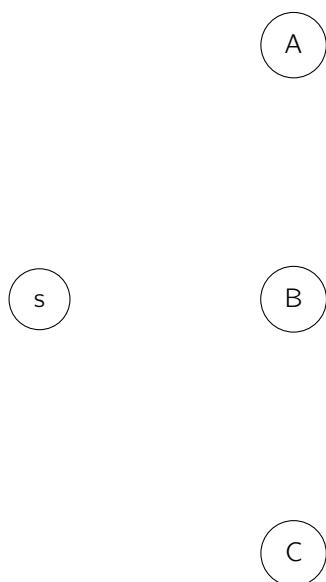
Schritt 6:

Nächstes Flussnetzwerk mit aktuellem Fluss:



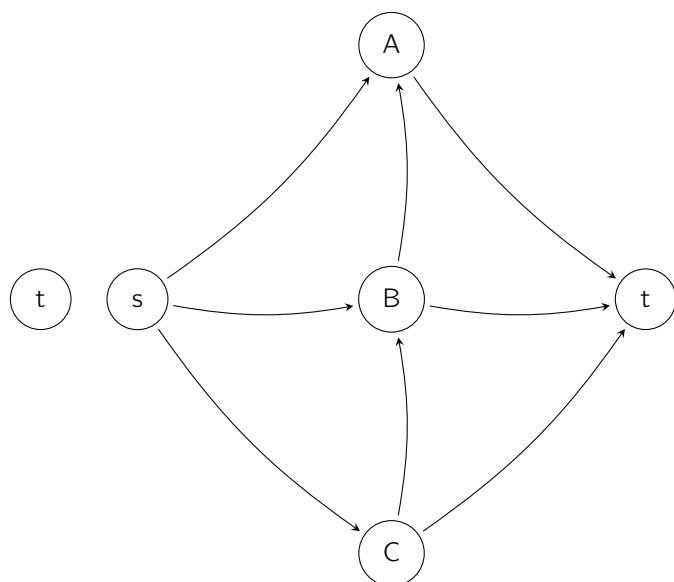
Schritt 7:

Restnetzwerk:



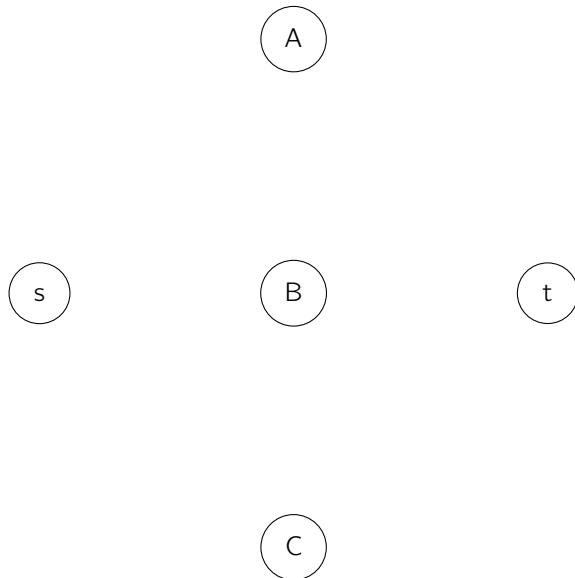
Schritt 8:

Nächstes Flussnetzwerk mit aktuellem Fluss:



Schritt 9:

Restnetzwerk:

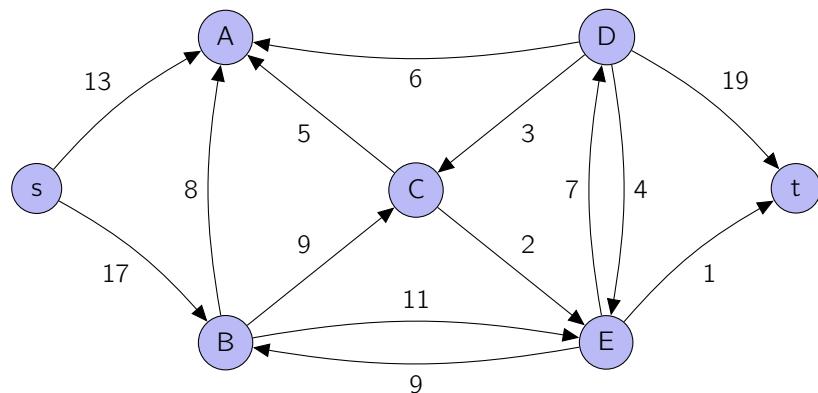


Der maximale Fluss hat den Wert:

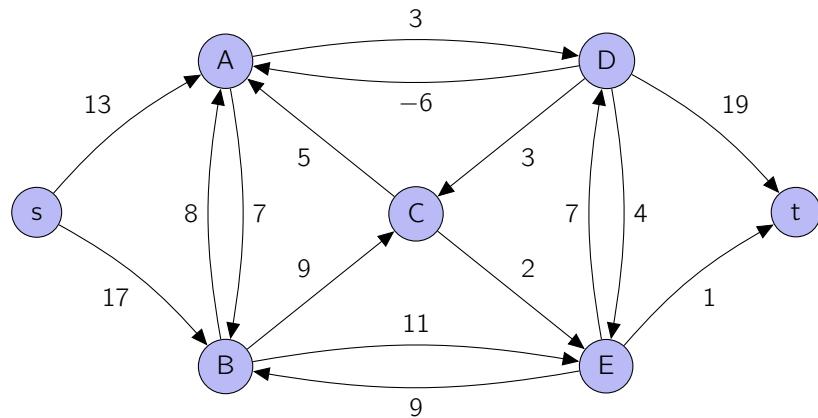
### Tutoraufgabe 3 (Flussnetzwerke):

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

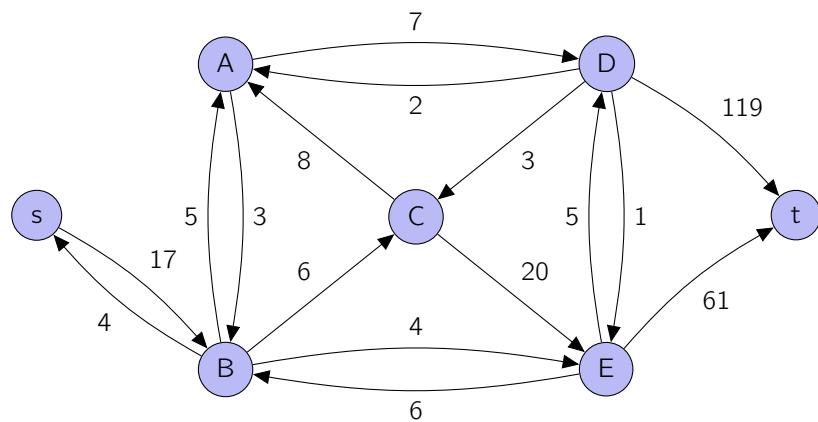
a)



b)



c)



#### 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:

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

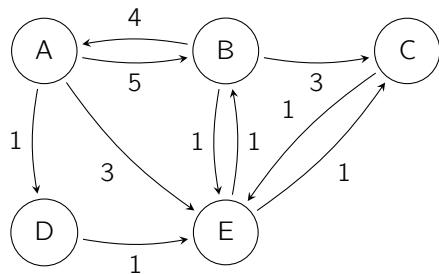
d) Sei  $(S, T)$  ein Schnitt über  $G$ , so gilt:

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

### 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 Entfernungswerte 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  $\infty$ , 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	$\infty$	1	3
B	4	0	3	$\infty$	1
C	$\infty$	$\infty$	0	$\infty$	1
D	$\infty$	$\infty$	$\infty$	0	1
E	$\infty$	1	1	$\infty$	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					

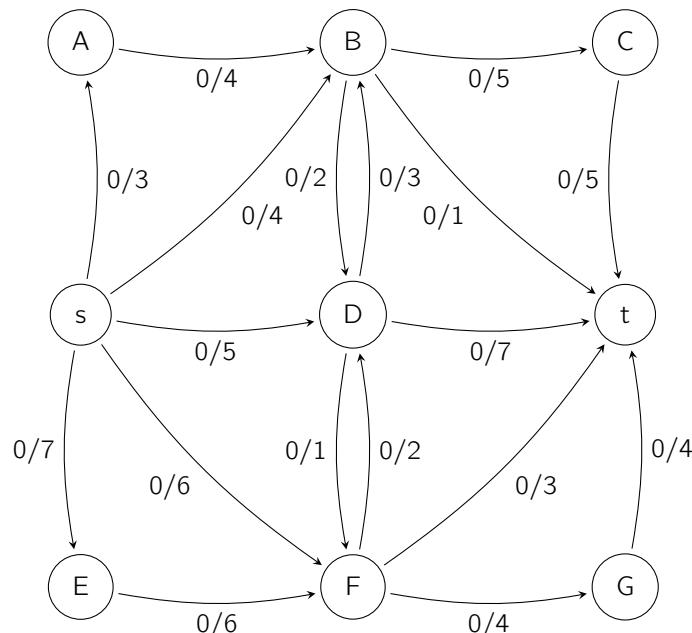
(5)	A	B	C	D	E
A					
B					
C					
D					
E					

(6)	A	B	C	D	E
A					
B					
C					
D					
E					

### Aufgabe 6 (Ford-Fulkerson):

**(9 Punkte)**

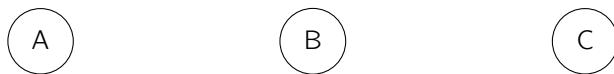
Betrachten Sie das folgende Flussnetzwerk mit Quelle s und Senke t:



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.

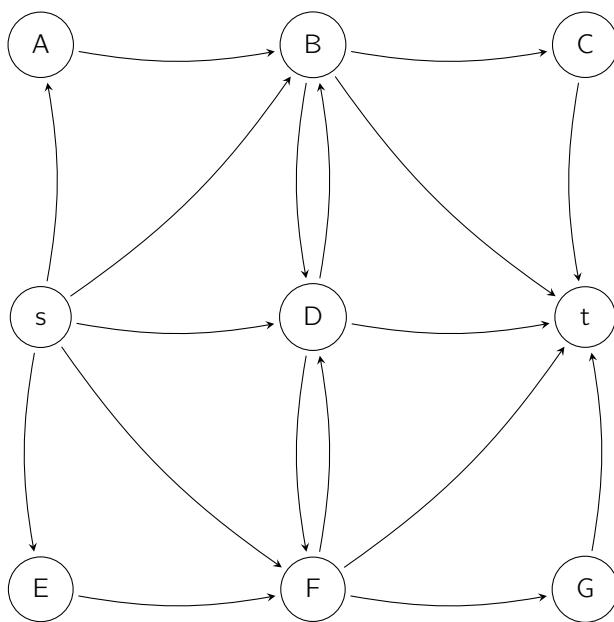
Schritt 1:

Restnetzwerk:



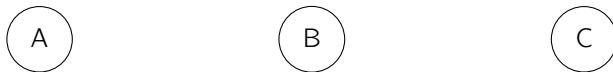
Schritt 2:

Nächstes Flussnetzwerk mit aktuellem Fluss:



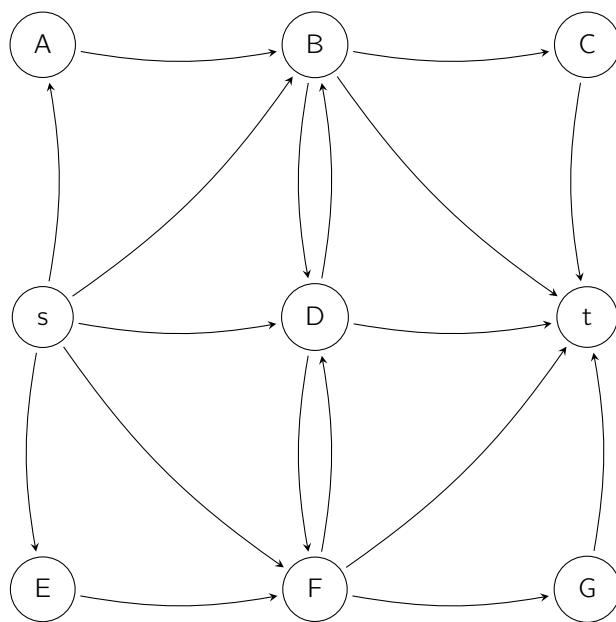
Schritt 3:

Restnetzwerk:



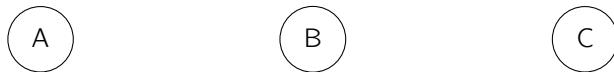
Schritt 4:

Nächstes Flussnetzwerk mit aktuellem Fluss:



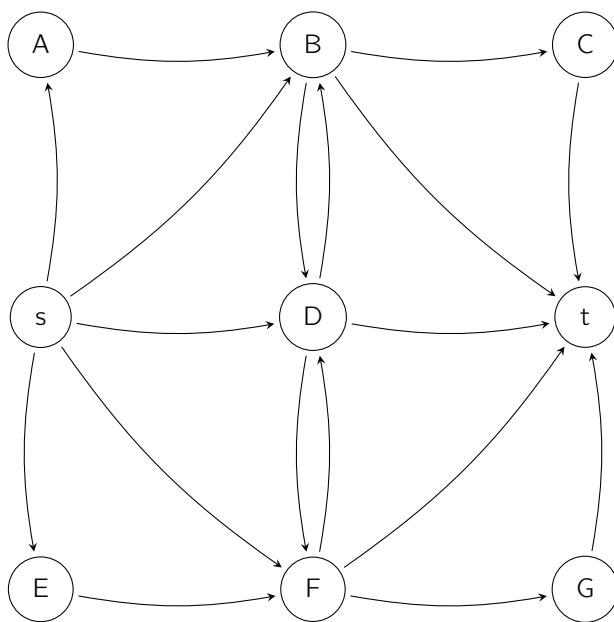
Schritt 5:

Restnetzwerk:



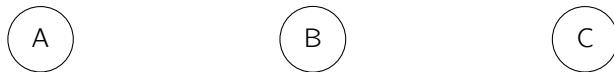
Schritt 6:

Nächstes Flussnetzwerk mit aktuellem Fluss:



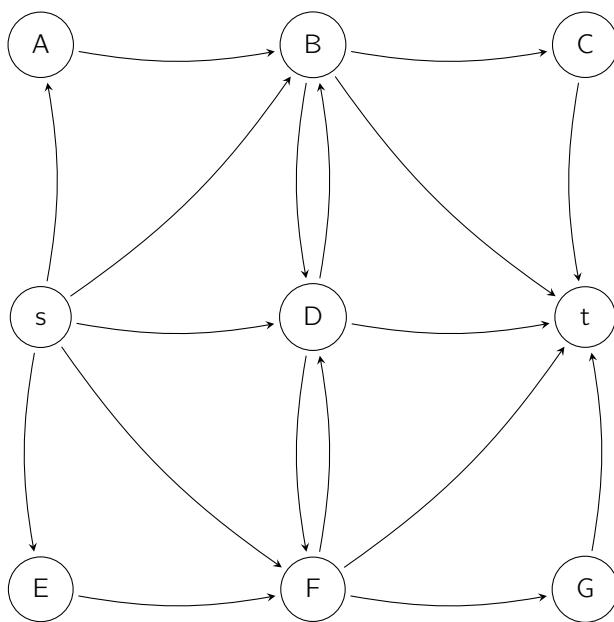
Schritt 7:

Restnetzwerk:



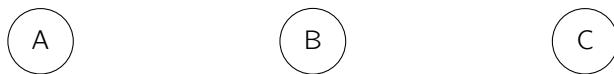
Schritt 8:

Nächstes Flussnetzwerk mit aktuellem Fluss:



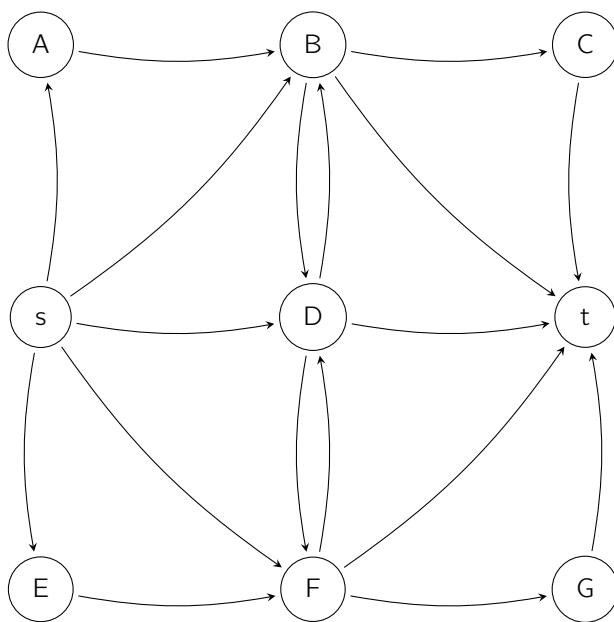
Schritt 9:

Restnetzwerk:



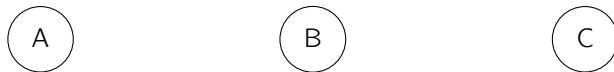
Schritt 10:

Nächstes Flussnetzwerk mit aktuellem Fluss:



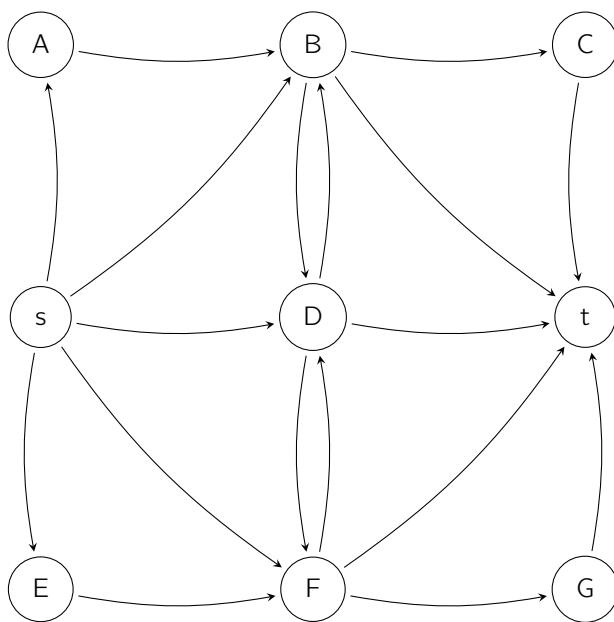
Schritt 11:

Restnetzwerk:



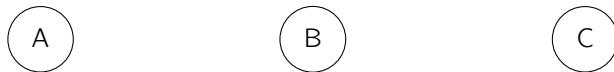
Schritt 12:

Nächstes Flussnetzwerk mit aktuellem Fluss:



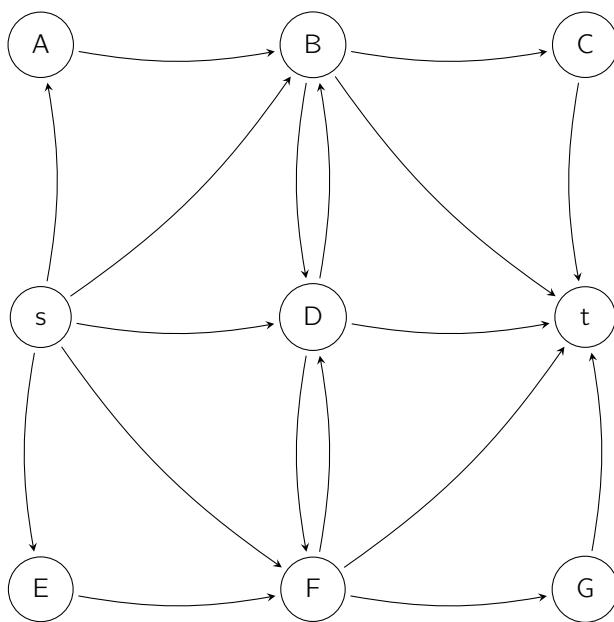
Schritt 13:

Restnetzwerk:



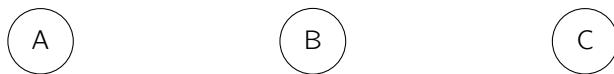
Schritt 14:

Nächstes Flussnetzwerk mit aktuellem Fluss:



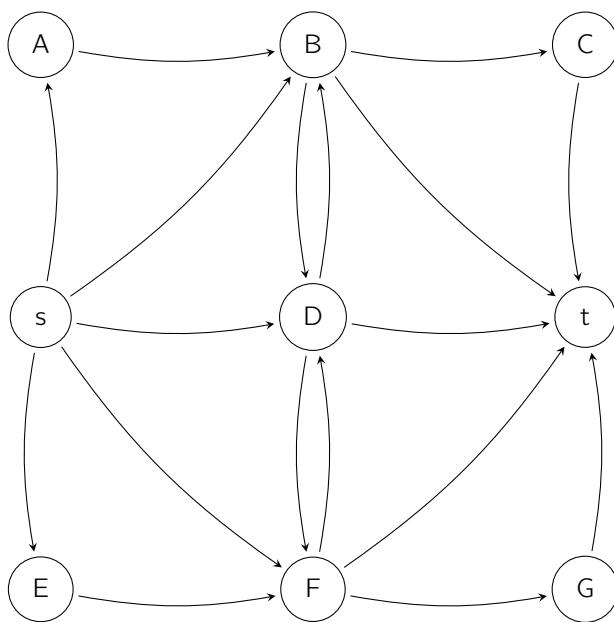
Schritt 15:

Restnetzwerk:



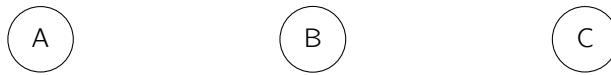
Schritt 16:

Nächstes Flussnetzwerk mit aktuellem Fluss:



Schritt 17:

Restnetzwerk:



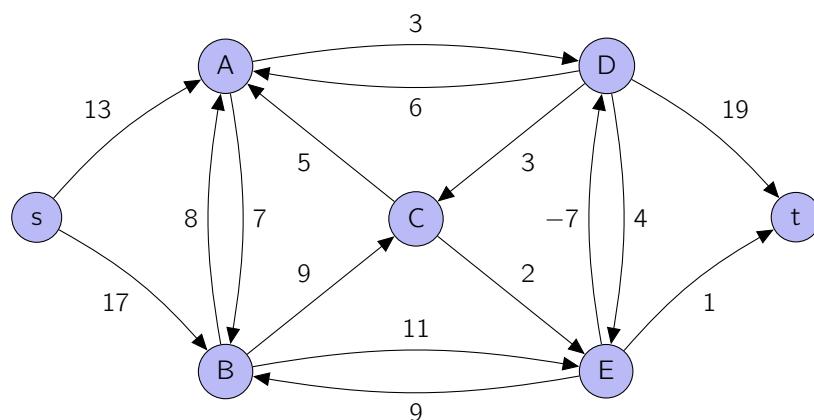
Der maximale Fluss hat den Wert:

**Aufgabe 7 (Flussnetzwerke):**

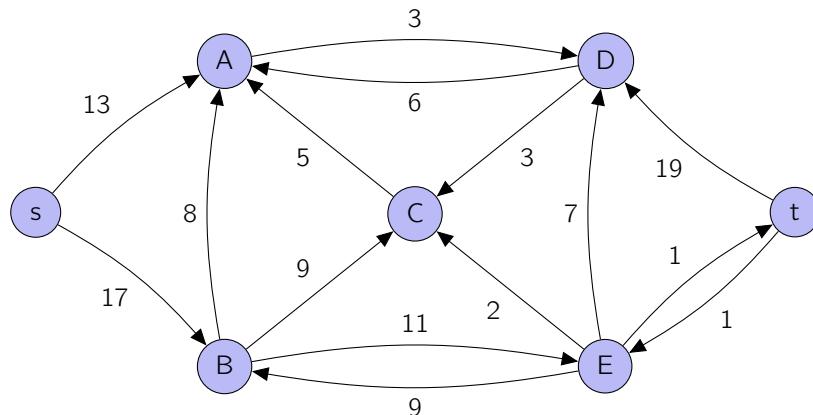
**(2 + 2 + 2 Punkte)**

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

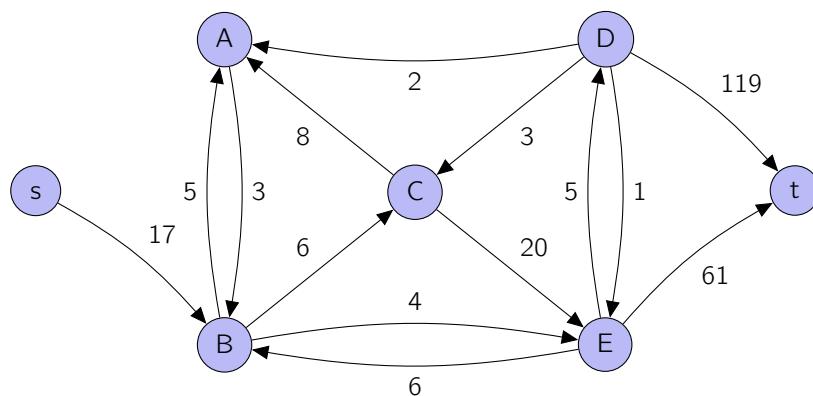
a)



b)



c)



### Aufgabe 8 (Flussnetzwerke und Flüsse):

(10 Punkte)

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

$$G = (V, E)$$

$$V = \mathbb{N} \cup \{\infty\}$$

$$E = \{(x, y) \mid x < y\}$$

zusammen mit der Kapazitätsfunktion

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

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

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

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.

**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