

Hinweise:

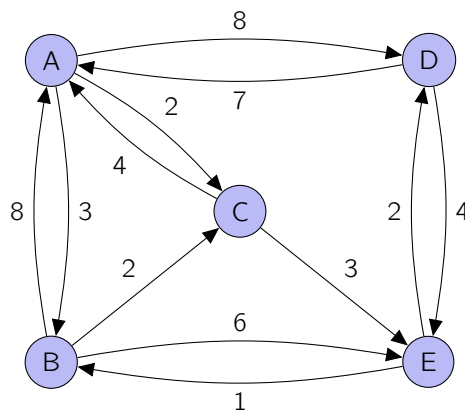
- Die Übungsblätter sind in Gruppen von je 3 Studierenden der gleichen Kleingruppenübung zu bearbeiten.
- Die Lösungen müssen bis Montag, den 19. Juli um 11:00 Uhr in die Übungskasten eingeworfen werden.
- Namen und Matrikelnummern, sowie die Nummer der Übungsgruppe sind auf jedes Blatt zu schreiben.

Aufgabe 1 (Floyd-Algorithmus):

(8 Punkte)

Bestimmen Sie, für den folgenden Graphen, die Werte der kürzesten Pfade zwischen allen Paaren von Knoten. Nutzen Sie hierzu den Algorithmus von Floyd. Geben Sie das Distanzarray vor dem Aufruf, sowie nach jeder Iteration an.

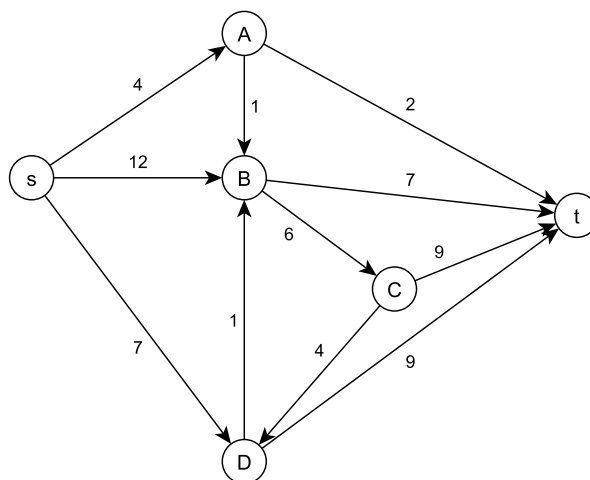
Die Knotenreihenfolge sei mit A, B, C, D, E fest vorgegeben.



Aufgabe 2 (Maximaler Fluss):

(8 Punkte)

Verwenden Sie die Ford-Fulkerson-Methode, um den maximalen Fluss des folgenden Flussnetzwerkes mit den gegebenen Kapazitäten zu berechnen. Geben Sie nach jeder Flusserhöhung das Flussnetzwerk an und kennzeichnen Sie den augmentierten Pfad. Geben Sie ebenfalls das Restnetzwerk nach der ersten, zweiten und letzten Flusserhöhung an.



Aufgabe 3 (Eigenschaften von Flüssen):

(2+3+4+2 Punkte)

Beweisen Sie die folgenden Aussagen über einen Fluss f und 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 4 (Ausgabe des kürzesten Pfades):

(5+3 Punkte)

- a) Erweitern Sie die in der Vorlesung gegebene Implementierung von Floyd (Vorlesung 18, Folie 21), sodass jeweils die zugehörigen Vorgänger in einem zusätzlichen Array `int &V[] []` abgelegt werden (vgl. Folie 22). Ihre Methode soll die folgende Signatur besitzen:

```
void floydSP(double W[] [], int n, double &D[] [], int &V[] [])
```

- b) Geben Sie eine Implementierung der Methode `void pfadAusgeben(int &V[] [], int start, int ziel)` an, die auf Basis des von der Methode aus **a)** berechneten Vorgängerarrays den kürzesten Pfad zwischen `start`- und `ziel`-Knoten ausgibt.