ADS 2 Klausur 2020

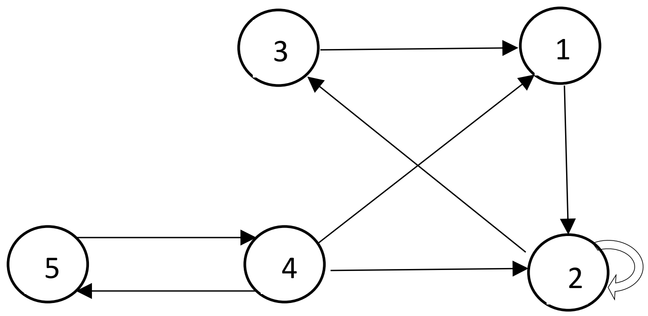
- Gedächtnisprotokoll -

Zeit: 60 min | online +10 min | offline +5 min

Die erste Klausur fand online statt. Die Nachklausur war sehr ähnlich zu der ersten, jedoch fand diese wieder in Präsenz statt. Der einzige Unterschied bestand darin, dass man bei der Nachklausur wieder schriftlich lösen musste. Die Aufgabenstellungen waren sonst nicht schwerer oder leichter gestaltet und waren teilweise identisch mit der Erstklausur.

1. Aufgabe

Graph G:



Adjazenzmatrix für G angeben:

0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	1	0	0	1
0	0	0	1	0

2. Aufgabe

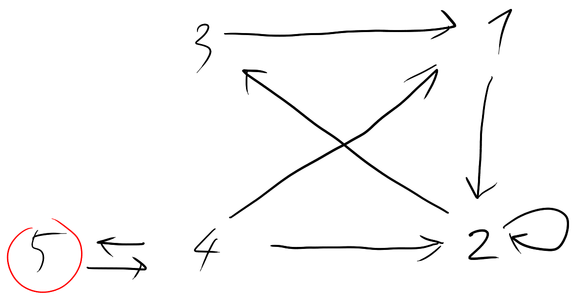
Konvertieren der Adjazenzmatrix in eine Kantenliste:

5, 8, 1,2, 2,2, 2,3, 3,1, 4,1, 4,2, 4,5, 5,4

3. Aufgabe

Tarjan-Algorithmus ab Knoten 5.
Knoten werden in aufsteigender Reihenfolge {1, 2, 3, 4, 5} bearbeitet.

v	in[v]	l[v]
3	5	3
2	4	3
1	3	3
4	2	1
5	1	1



4. Aufgabe

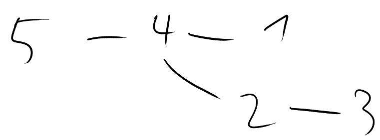
Reihenfolge der Komponenten {1, 2, 3} und {4,5}, in der sie der Algorithmus während der Abarbeitung ausgibt, angeben:

{3, 2, 1}   {4, 5}

5. Aufgabe

Breitensuchdurchlauf in aufsteigender Reihenfolge ab Knoten 5 durchführen.

{5, 4, 1, 2, 3}



6. Aufgabe

Bestimmen der transitiven Hülle (Naiver Ansatz oder Warshall Algorithmus)

1	1	1	0	0
1	1	1	0	0
1	1	1	0	0
1	1	1	1	1
1	1	1	1	1

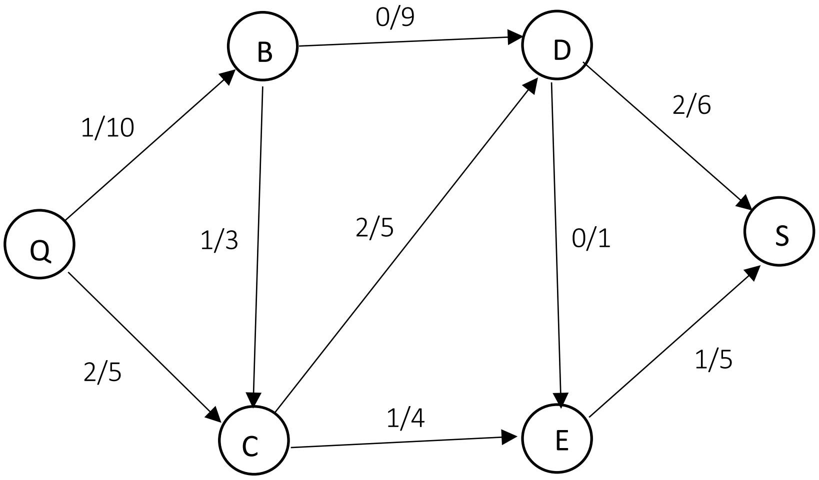
7. Aufgabe

Die transitive Hülle entspricht nicht der reflexiven Hülle

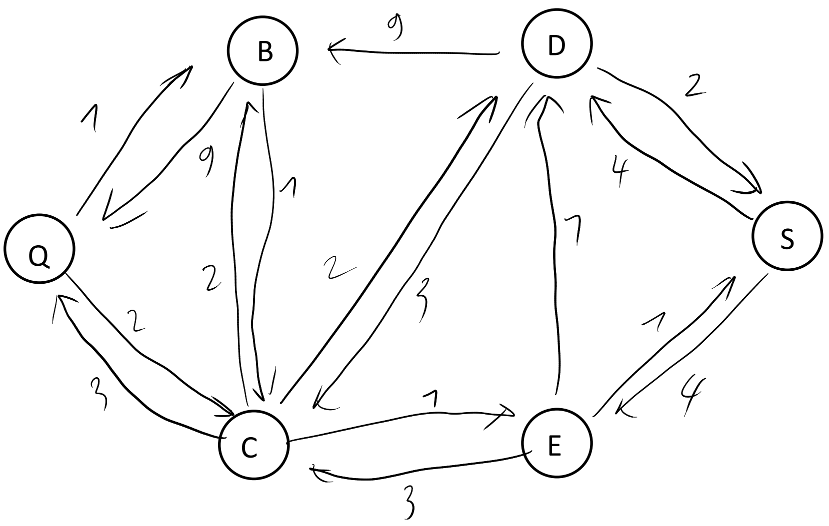
- ○ wahr ? ○ falsch

8. Aufgabe

Gegeben ist folgendes Flussnetzwerk:



Hierzu Restgraphen erzeugen:



9. Aufgabe

Maximal möglichen Fluss angeben, falls gültiger Erweiterungspfad. Wenn nicht gültig, 0 angeben.

a) (Q, B, D, S) 4

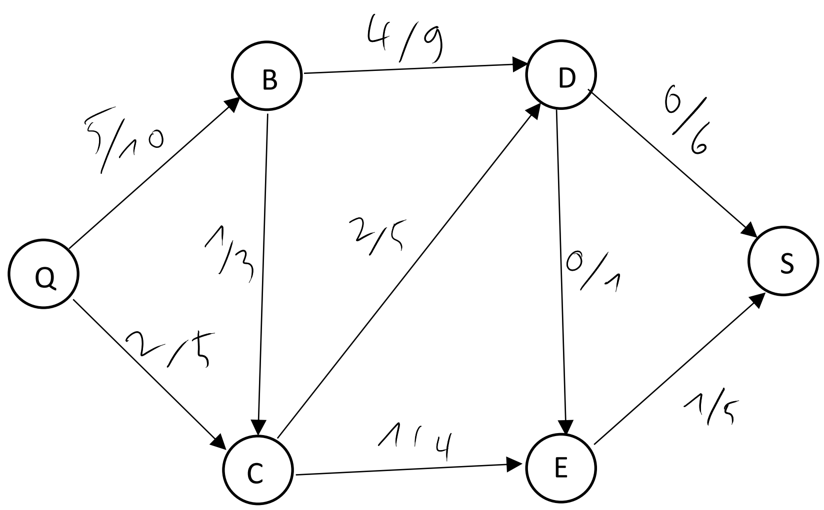
b) (Q, C, V, D, C, E, S) 0

c) (Q, C, E, S) 3

d) (Q, B, D, C, E, S) 1

10. Frage

Erhöhung des Flusses entlang des Erweiterungspfades mit minimalen Extrafluss. Hierzu einen Pfad aus der Aufgabe 9 wählen.



11. Aufgabe

Distanzmatrix, mit den Städten (1,2,3,4):

$g(a,b)$: $a \downarrow$, $\rightarrow b$

	1	2	3	4
1	0	4	3	6
2	5	0	1	2
3	2	2	0	7
4	4	6	4	0

Hier wird die längste Rundreise gesucht (Traveling Salesman Problem). Jede Stadt wird genau einmal besucht, bevor der Ausgangspunkt wieder erreicht wird.
(Hier stand noch die Formel für TSP)

Vervollständigen folgender Formel:

g(3,{2,4}) = $\max\left(d_{3,2} + g(2,\{4\}),\ d_{3,4} + g(4,\{2\})\right)$

12. Aufgabe

Alle Zwischenwerte angeben bzw. Werte nach dem aus den Übungen bekannten Schema berechnen:

S = ∅
g(2, {}) = 5
g(3, {}) = 2
g(4, {}) = 4

|S| = 1:
g(2, {3}) = 1 + g(3, {}) = 1 + 2 = 3
g(3, {2}) = 2 + 5 = 7
g(2, {4}) = 2 + 4 = 6
g(4, {2}) = 6 + 5 = 11
g(3, {4}) = 7 + 4 = 11
g(4, {3}) = 4 + 2 = 6

|S| = 2:
g(2,{3,4}) = $\max(1 + \overset{g(3,4)}{11},\ 2 + \overset{g(4,3)}{6}) = 1 + 11 = 12$
g(3,{2,4}) = $\max(2 + 6,\ 7 + 11) = 7 + 11 = 18$
g(4,{2,3}) = $\max(6 + 3,\ 4 + 7) = 4 + 11 = 11$

Länge einer längsten Rundreise g(1,{2,3,4}) = $\max(5 + \overset{g(2,\{3,4\})}{12},\ 2 + \overset{g(3,\{2,4\})}{18},\ 4 + \overset{g(4,\{2,3\})}{11})$

$= 2 + 18 = 20$

1 3 4 2 1

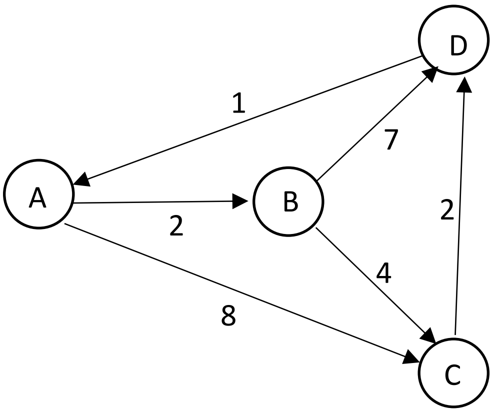
13. Aufgabe

Eine längste Rundreise:

Start: 1, 4, 3, 2, 1, Ende.

14. Aufgabe

Gerichteter Graph:

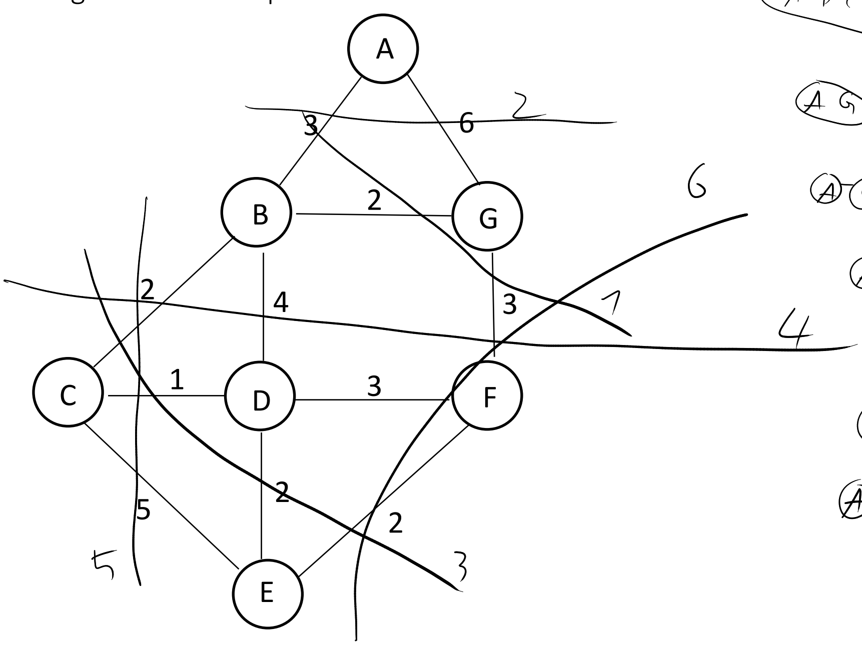


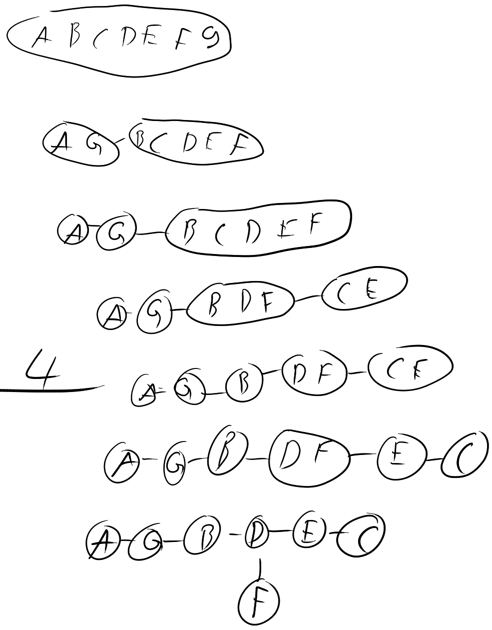
Dijkstra-Algorithmus anwenden, um die Längen der kürzesten Pfade von Knoten A zu allen anderen Knoten zu berechnen.

	D[A]	D[B]	D[C]	D[D]
A	0	*	*	*
A	0	2	8	*
B	0	2	6	9
C	0	2	6	8
D	0	2	6	8

15.Aufgabe

Ungerichteter Graph:



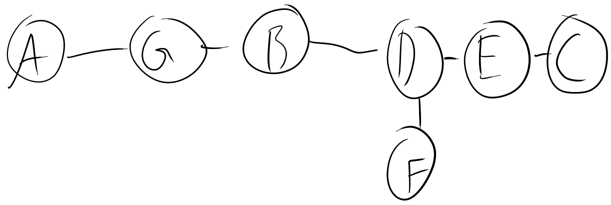


Gomory-Hu Algorithmus anwenden:

V	S	t	min-cut(s,t)
{A, B, C, D, E, F, G}	A	B	8
{A, G}, {B, C, D, E, F}	A	G	9
{A}{G}, {B, C, D, E, F}	B	C	7
{A}{G}{B, D, F}{C, E}	B	D	9
{A}{G}{B}{D, F}{C, E}	C	E	8
{A}{G}{B}{D, F}{C}{E}	D	F	8

16. Aufgabe

Gomory-Hu Baum zeichnen:



17. Aufgabe

Optimale Alignement für die Strings S1= DABC und S2= AAC mittels Hirschberg Algorithmus berechnen. Rekursion auf S1 anwenden.

D A B C

↓ ↓ ↓

\- A A C

	-	D	A				
-	0	1	2	2	2	3	A
A	1	1	1	1	1	2	A
A	2	2	1	1	0	1	C
C	3	3	2	2	1	0	-
				B	C	-	

	-	B			
-	0	1	1	2	A
A	1	1	0	1	C
C	2	2	1	0	-
			C	-	

18. Aufgabe

Berechnetes Alignement der beiden Strings:

S1 = 1

S2 = 1

19. Aufgabe

für alle "a" muss gelten ggT(a,n) = 1
(ggT = größter gemeinsamer Teiler)

Multiplikationstabelle für die Gruppe (Z_8^*, $\times_8$) $\{u \in \mathbb{Z}_n : ggT(u,n) = 1\}$

$1, \cancel{2}, 3, \cancel{4}, 5, \cancel{6}, 7, \cancel{8}$

$\times_8$	1	3	5	7
1	1	3	5	7
3	3	1	7	5
5	5	7	1	3
7	7	5	3	1

20. Aufgabe

Primfaktor von n = 9 mit Hilfe von Pollards Rho-Algorithmus finden: $x_1 = 4$ $k = 2$

1	x_{i-1}^2 - 1	x_i	d	y	
Start i=1 2	-	4 $= x_1$	-	4	
i=1+1 3	$16 - 1 = 15$	$x_2 = 15 \% 9 = 6$	$(x_2 - y, n)$ ggT $(6-4, 9) = 1$	$6 = x_i$	$\rightarrow l = 2k$ $k = 4$
i=2+1 4	$6^2 = 36 - 1 = 35$	$35 \% 9 = 8 = x_3$	$(x_3 - y, n)$ ggT $(8-6, 9)$ ggT $= 1$	6	
5	$8^2 - 1 = 63$	$63 \% 9 = 0 = x_4$	$(0 - 6, 9)$ ggT $= 3$	0	$l = 8$ Print(3)
6					

nach berechnung von d:

```
if (d != 1 and d != n):
    return d; #programm beendet

if (i==k):
    y = xi
    k = 2k
```

Der ermittelte Primfaktor: 3