Algorithmen und Datenstrukturen - Hausübung 11

Gruppenmitglieder

- Emre Berber (2957148)
- Christoph Berst (2743394)
- Jan Braun (2768531)

Inhaltsverzeichnis

H1

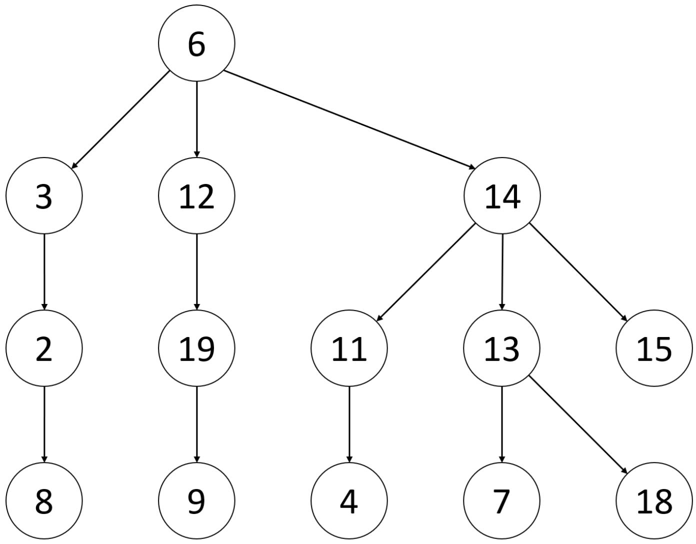
a)

i)

Iteration	u	v	Q
0	–	$\square$	{6}
1	6	[3,12,14]	{3,12,14}
2	3	[2]	{12,14,2}
3	12	[19]	{14,2,19}
4	14	[11,13,15]	{2,19,11,13,15}
5	2	[8]	{19,11,13,15,8}
6	19	[9]	{11,13,15,8,9}
7	11	[4]	{13,15,8,9,4}
8	13	[7,18]	{15,8,9,4,7,18}
9	15	$\square$	{8,9,4,7,18}
10	8	$\square$	{9,4,7,18}
11	9	$\square$	{4,7,18}
12	4	$\square$	{7,18}
13	7	$\square$	{18}
14	18	$\square$	$\emptyset$
15	–	$\square$	$\emptyset$
16	–	$\square$	$\emptyset$

ii)

BFS-Baum G^6_{pred}



b)

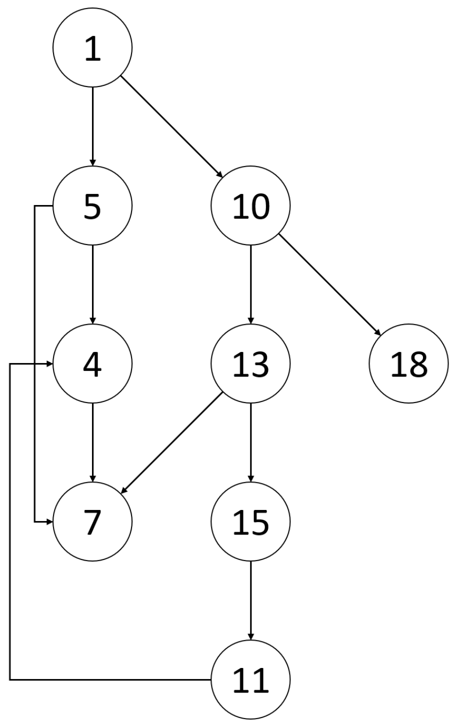
i)

Knoten	Entdeckungszeit	Abschlusszeit	Vorgängerknoten
1	1	18	nil
2	19	20	nil
3	21	22	nil
4	3	6	5
5	2	7	1
6	23	24	nil
7	4	5	4
8	25	26	nil
9	27	28	nil
10	8	17	1
11	11	12	15
12	29	30	nil
13	9	16	10
14	31	32	nil
15	10	13	13
16	33	34	nil
17	35	36	nil
18	14	15	13
19	37	38	nil
20	39	40	nil

ii)

topologisch Sortieren:
1,10,13,18,15,15,11,5,4,7

DFS-Wald G_{pred}

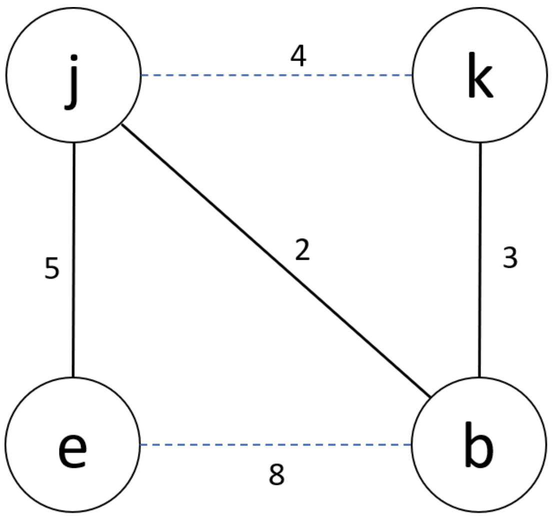


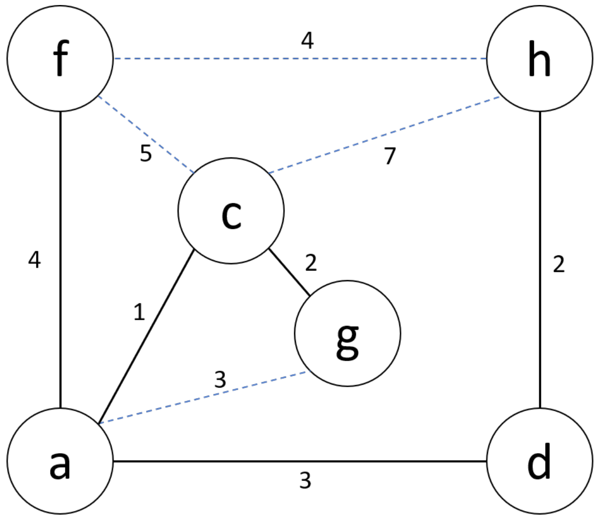
H2

Für die Zeichnungen in dieser Aufgabe gilt, dass gestrichelten Linien sollen nicht Teil des Spannbaums sein.

a)

$\{u,v\}$	$w(\{u,v\})$	Dazu?	set(a)	set(b)	set(c)	set(d)	set(e)	set(f)	set(g)	set(h)	set(j)	set(k)
□	□	□	{a}	{b}	{c}	{d}	{e}	{f}	{g}	{h}	{j}	{k}
$\{a,c\}$	1	true	{a,c}	=	{a,c}	=	=	=	=	=	=	=
$\{b,j\}$	2	true	=	{b,j}	=	=	=	=	=	=	{b,j}	=
$\{c,g\}$	2	true	{a,c,g}	=	{a,c,g}	=	=	=	{a,c,g}	=	=	=
$\{d,h\}$	2	true	=	=	=	{d,h}	=	=	=	{d,h}	=	=
$\{a,d\}$	3	true	{a,c,d,g,h}	=	{a,c,d,g,h}		=	=	{a,c,d,g,h}		=	=
$\{a,g\}$	3	false	=	=	=	=	=	=	=	=	=	=
$\{b,k\}$	3	true	=	{b,j,k}	=	=	=	=	=	=	{b,j,k}	
$\{a,f\}$	4	true	{a,c,d,f,g,h}	=	{a,c,d,f,g,h}		=	{a,c,d,f,g,h}			=	=
$\{f,h\}$	4	false	=	=	=	=	=	=	=	=	=	=
$\{j,k\}$	4	false	=	=	=	=	=	=	=	=	=	=
$\{c,f\}$	5	false	=	=	=	=	=	=	=	=	=	=
$\{e,j\}$	5	true	=	{b,e,j,k}	=	=	{b,e,j,k}	=	=	=	{b,e,j,k}	
$\{c,h\}$	7	false	=	=	=	=	=	=	=	=	=	=
$\{b,e\}$	8	false	=	=	=	=	=	=	=	=	=	=

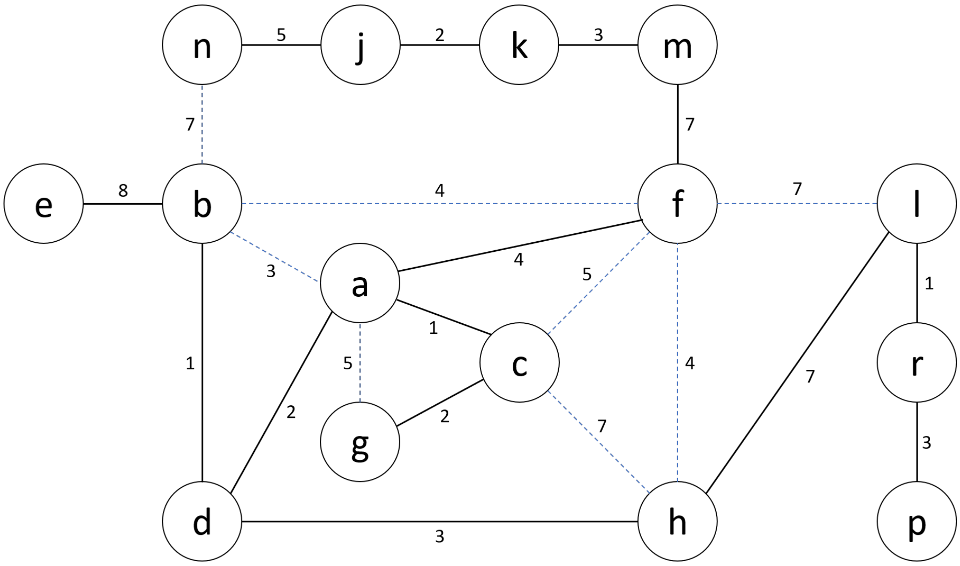




b)

a.k	b.k	c.k	d.k	e.k	f.k	g.k	h.k	j.k	k.k	l.k	m.k	n.k	p.k	r.k	u	Q
$-\infty$	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	–	$\{a,b,c,d,e,f,g,h,j,k,l,m,n,p,r\}$
$-\infty$	3	1	2	∞	4	5	∞	∞	∞	∞	∞	∞	∞	∞	a	$\{c,d,b,f,g,e,h,j,k,l,m,n,p,r\}$
$-\infty$	3	1	2	∞	4	2	7	∞	∞	∞	∞	∞	∞	∞	c	$\{d,g,b,f,h,e,j,k,l,m,n,p,r\}$
$-\infty$	1	1	2	∞	4	2	3	∞	∞	∞	∞	∞	∞	∞	d	$\{b,g,h,f,e,j,k,l,m,n,p,r\}$
$-\infty$	1	1	2	8	4	2	3	∞	∞	∞	∞	7	∞	∞	b	$\{g,h,f,n,e,j,k,l,m,p,r\}$
$-\infty$	1	1	2	8	4	2	3	∞	∞	∞	∞	7	∞	∞	g	$\{h,f,n,e,j,k,l,m,p,r\}$
$-\infty$	1	1	2	8	4	2	3	∞	∞	7	∞	7	∞	∞	h	$\{f,l,n,e,j,k,m,p,r\}$
$-\infty$	1	1	2	8	4	2	3	∞	∞	7	7	7	∞	∞	f	$\{l,m,n,e,j,k,p,r\}$
$-\infty$	1	1	2	8	4	2	3	∞	∞	7	7	7	∞	1	l	$\{r,m,n,e,j,k,p\}$
$-\infty$	1	1	2	8	4	2	3	∞	∞	7	7	7	3	1	r	$\{p,m,n,e,j,k,\}$
$-\infty$	1	1	2	8	4	2	3	∞	∞	7	7	7	3	1	p	$\{m,n,e,j,k,\}$
$-\infty$	1	1	2	8	4	2	3	∞	3	7	7	7	3	1	m	$\{k,n,e,j\}$
$-\infty$	1	1	2	8	4	2	3	2	3	7	7	7	3	1	k	$\{j,n,e\}$
$-\infty$	1	1	2	8	4	2	3	2	3	7	7	5	3	1	j	$\{n,e\}$
$-\infty$	1	1	2	8	4	2	3	2	3	7	7	5	3	1	n	$\{e\}$
$-\infty$	1	1	2	8	4	2	3	2	3	7	7	5	3	1	e	$\emptyset$

a.p	b.p	c.p	d.p	e.p	f.p	g.p	h.p	j.p	k.p	l.p	m.p	n.p	p.p	r.p	u	Q
nil	nil	nil	nil	nil	nil	nil	nil	nil	nil	nil	nil	nil	nil	nil	–	$\{a,b,c,d,e,f,g,h,j,k,l,m,n,p,r\}$
nil	a	a	a	nil	a	a	nil	nil	nil	nil	nil	nil	nil	nil	a	$\{c,d,b,f,g,e,h,j,k,l,m,n,p,r\}$
nil	a	a	a	nil	a	c	c	nil	nil	nil	nil	nil	nil	nil	c	$\{d,g,b,f,h,e,j,k,l,m,n,p,r\}$
nil	d	a	a	nil	a	c	d	nil	nil	nil	nil	nil	nil	nil	d	$\{b,g,h,f,e,j,k,l,m,n,p,r\}$
nil	d	a	a	b	a	c	d	nil	nil	nil	nil	b	nil	nil	b	$\{g,h,f,n,e,j,k,l,m,p,r\}$
nil	d	a	a	b	a	c	d	nil	nil	nil	nil	b	nil	nil	g	$\{h,f,n,e,j,k,l,m,p,r\}$
nil	d	a	a	b	a	c	d	nil	nil	h	nil	b	nil	nil	h	$\{f,l,n,e,j,k,m,p,r\}$
nil	d	a	a	b	a	c	d	nil	nil	h	f	b	nil	nil	f	$\{n,l,m,e,j,k,p,r\}$
nil	d	a	a	b	a	c	d	nil	nil	h	f	b	nil	l	l	$\{r,m,n,e,j,k,p\}$
nil	d	a	a	b	a	c	d	nil	nil	h	f	b	r	l	r	$\{p,m,n,e,j,k\}$
nil	d	a	a	b	a	c	d	nil	nil	h	f	b	r	l	p	$\{m,n,e,j,k\}$
nil	d	a	a	b	a	c	d	nil	m	h	f	b	r	l	m	$\{k,n,e,j\}$
nil	d	a	a	b	a	c	d	k	m	h	f	b	r	l	k	$\{j,n,e\}$
nil	d	a	a	b	a	c	d	k	m	h	f	j	r	l	j	$\{n,e\}$
nil	d	a	a	b	a	c	d	k	m	h	f	j	r	l	n	$\{e\}$
nil	d	a	a	b	a	c	d	k	m	h	f	j	r	l	e	$\emptyset$



H3

a)

Induktionsanfang: r = 1
$a^1_{i,j} = 1$ falls $\{v_i, v_j\} \in E$. Gleichzeitig ist die Anzahl der Wege zwischen v_i und v_j auch 1, da es nur diese direkte Verbindung von v_i zu v_j gibt. Andernfalls würde keine Kante existieren und damit wäre $a^1_{i,j} = 0$.

Induktionsvorraussetzung:
Für alle i, j ist der Eintrag $a^{(r)}_{i,j} = (A^r)_{i,j}$ gleich der Anzahl der Wege der Länge r in G zwischen den Knoten i und j. Weiterhin gilt falls kein Weg der Länge r existiert, welcher v_i und v_j verbindet, steht in $a^{(r)}_{i,j} = (A^r)_{i,j} = 0$

Induktionsschritt: r → r+1
Falls es keinen Weg der Länge r+1 zwischen v_i und v_j gibt, steht an der $a^{(r)}_{i,j} = 0$. Gibt es einen Weg zwischen v_i und v_j, dann steht in $a^{(r+1)}_{i,j}$ die Anzahl der paarweise verschiedenen Wege zwischen v_i und v_j der Länge r+1.

b)

```
GraphTriangles(G,V,E)
    c=0
    FOR u=0 TO V.length−1
        FOR v=u+1 TO adj(G,u).length−1
            IF adj(G,u,V[v]) == 0
                CONTINUE
            FOR w=v+1 TO adj(G,u).length−1
                IF adj(G,u,V[w]) == 0
                    CONTINUE
                IF {V[v],V[w]} ∈ E
                    c++
    return c
```

- V ist die Liste der Knoten in G und E ist die Liste der Kanten in G. {v,w} ∈ E ist eine ungerichtete Kante zwischen v und w.
- adj(G,v,w) gibt das Gewicht der Kante zwischen v und w zurück. Mit v,w ∈ V
- adj(G,u).length−1 sollte identisch zu V.length−1 sein, da es sich bei adj(G,u) im die Zeile des Knotens u in der Adjazenzmatrize von G handelt, welche mit dem Wert 0 angibt das keine Verbindung von u zu V[i] besteht und ansonsten das Gewicht der Kante zwischen den Knoten u und V[i] enthält.

c)

Komplexität:
Wir druchlaufen n Knoten mit u, maximal n Kanten, in denen u enthalten ist, mit v und maximal weitere n Kanten, in denen auch u enthalten ist, mit w. Dies macht dann n $\cdot$ n $\cdot$ n = n^3.

Korrektheit:
Invariante:
Alle Dreiecke bis zum u-ten Schritt, welche einen der u ersten Knoten enthalten, sind gezählt worden.

Initialisierung:
Solange kein Knoten betrachtet wurden, ist die Anzahl der Dreiecke auch einfach 0.

Fortsetzung:
Für die bisherigen Knoten sind alle Dreiecke gefunden worden, die einen von diesen enthalten. Dies ist der Fall, da wir uns immer nur die weiteren Knoten nach unserem aktuellen Knoten anschauen. Somit schauen wir uns sowohl keinen der bisherigen nochmal an, noch versuchen wir ein Dreieck mit mehrmals dem selben Knoten zu bilden.

Terminierung:
Die Schleife enden mit dem letzten Knoten V.length-1. Es müssten nun auch alle Dreiecke im Graphen gefundene worden sein.

Die selbe Argumentation kann man auf die inneren Schleifen übertragen.