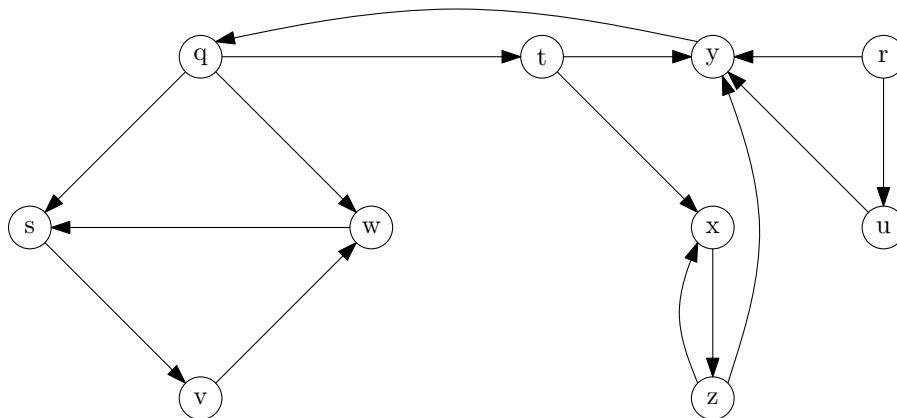


Aufgabe 1

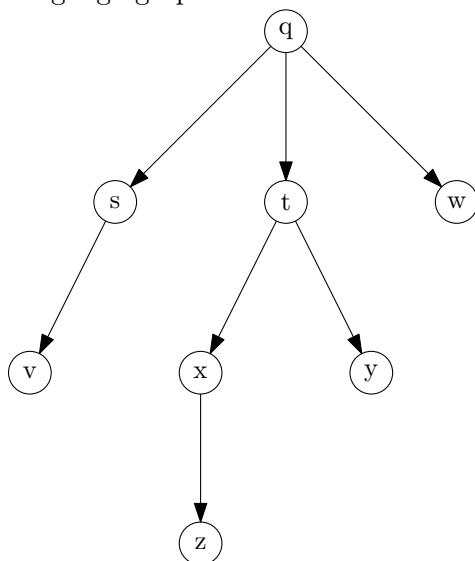
a)



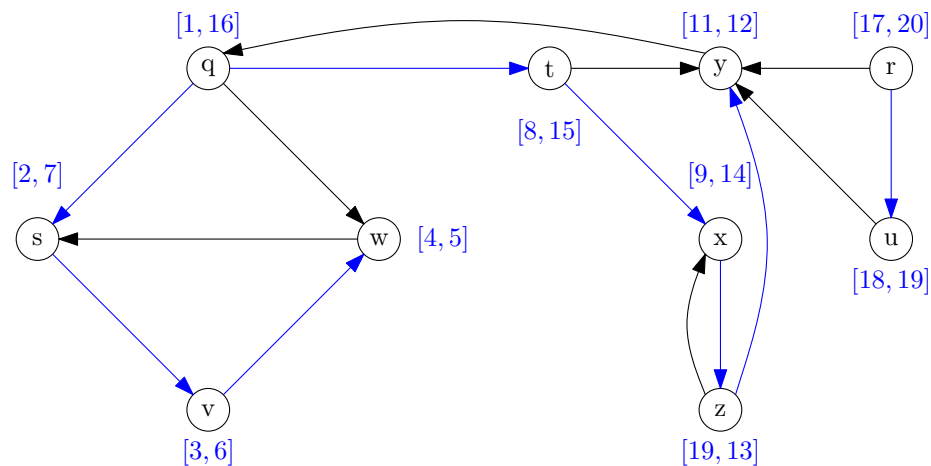
a	q	s	t	w	v	x	y	z	r	u
$d(a)$	0	1	1	1	2	2	2	3	∞	∞
$\pi(a)$	-	q	q	q	s	t	t	x	-	-

b)

Vorgängergraph:



c)



Aufgabe 2

a)

Kante $(u, v) \in E$

Angenommen (u, v) ist eine TreeEdge (ForwardEdge)

\Rightarrow In der Tiefensuche wird v von u aus entdeckt.

$\Rightarrow v$ ist ein Nachkomme von u im Tiefensuchebaum

\Rightarrow Verschachtelungssatz:

$$d[u] < d[v] < f[v] < f[u]$$

' \Rightarrow Angenommen $d[u] < d[v] < f[v] < f[u]$

\Rightarrow Verschachtelungssatz:

v ist ein Nachkomme von u im selben Tiefensuchebaum.

Fall 1: v wurde von u aus entdeckt $\Rightarrow (u, v)$ TreeEdge

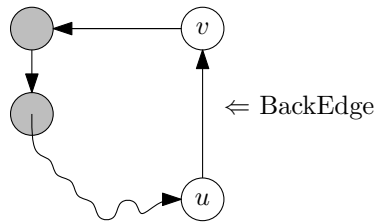
Fall 2: v wurde nicht von u aus entdeckt $\Rightarrow (u, v)$ ForwardEdge

b)

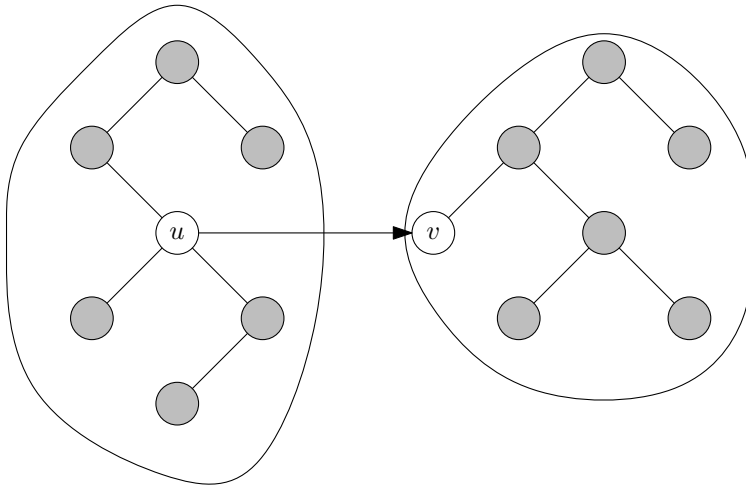
' \Rightarrow Angenommen (u, v) ist eine BackEdge

d. h. u ist ein Nachfolger von v im selben Tiefensuchebaum

\Leftrightarrow Verschachtelungssatz $d[v] < d[u] < f[u] < f[v]$



c)



' \Rightarrow ': (u, v) CrossEdge

Also sind u und v in verschiedenen Tiefensuchebäumen.

$\Rightarrow v$ muss vor u in der Tiefensuche entdeckt worden sein.

$\Rightarrow d[v] < f[v] < d[u] < f[u]$

\Leftarrow Angenommen: $d[v] < f[v] < d[u] < f[u]$

\Rightarrow Verschachtelungssatz: u und v sind in zwei verschiedenen TS-Bäumen

$\Rightarrow (u, v)$ ist CrossEdge

Aufgabe 3

Laufzeit: $O(\|V\|^2)$

sowohl Breitensuche und Tiefensuche

\Rightarrow Es müssen alle Spalten/Zeilen durchgegangen werden.

Aufgabe 4

Annahme:

Behauptung: Jede Tiefensuche liefert $d[v] < f[u]$ als Ergebnis.

Betrachte eine beliebige Tiefensuche zu dem Zeitpunkt, an dem v entdeckt wird.

Fall 1: u wurde vor v entdeckt

Weißer Pfadsatz: v ist ein Nachfolger von $u \rightarrow d[v] < f[u]$

Fall 2: u wurde noch nicht entdeckt:

a) $v \xrightarrow{P} u \Rightarrow u$ Nachkomme von v

Verschachtelungssatz: $d[v] < d[u] < f[u] < f[v]$

b) u ist von v aus nicht erreichbar.

Also wird v komplett abgearbeitet, bevor u entdeckt wird:

$d[v] < f[v] < d[u] < f[u]$

Aufgabe 5

Graph $G = (V, E)$ (gerichtet)

$E^T = \{(v, u) \mid (u, v) \in E\}$

```

1 MatrixTranspose(a):
2     n = size(a)
3     for i:= 1 to n do
4         for j:= 1 to n do
5             B[i][j] = A[j][i]
6     return B

```

Laufzeit: $\Theta(n^2)$

```

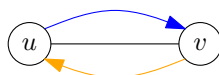
1 ListTranspose(G=(V,E)):
2     Erzeuge einen leeren Graphen G2 mit der Knotenmenge V
3     for jeden Knoten u in V do
4         for jeden Knoten v in Adj[u] do
5             Fuege Kante (v,u) zu G2 hinzu
6     return G2

```

Laufzeit: $O(\|V\| + \|E\|)$

Aufgabe 6

rekursiver Aufruf



Ende rekursiver Aufruf

Ansatz: Führe Tiefensuche durch und speichere die benutzten Kanten in einer Liste.

```

1 Search(G,s):
2     for each v in V do
3         color[v] := WHITE
4     Initialisiere die Liste p
5     Visit(G,s,p)

```

```
1 Visit(G,v,p):  
2   color[v] := GRAY  
3   Haenge v an das Ende von p  
4   for each u in Neighbour(v) do  
5     if color[u] = WHITE then  
6       Visit(G,u,p)  
7       Haenge v an das Ende von P  
8     else if color[u] = GRAY then  
9       Haenge (u,v) an das Ende von p  
10  color[v] := BLACK
```