Aufgabe 1. Gegeben ist der Graph G mit folgender Adjazenzlistendarstellung:

Knoten	Nachbarn
q	s, t, w
r	u, y
s	v
t	x, y
u	y
v	w
w	s
x	z
y	q
z	x, y

a) Stellen Sie den Graphen grafisch dar.

b) Führen Sie auf dem Graphen eine Breitensuche mit Startknoten q durch. Zeichnen Sie den berechneten Vorgängerteilgraphen.

c) Führen Sie auf dem Graphen eine Tiefensuche durch. Zeichnen Sie den von der Tiefensuche berechneten Depth-First Wald.

Hinweis: Wählen Sie die Knoten in der Reihenfolge aus, in der sie in obiger Tabelle eingetragen sind.

Aufgabe 2. Betrachten Sie das Ergebnis einer Tiefensuche auf einem Graphen $G = (V, E)$. Beweisen Sie: Eine Kante $(u, v) \in E$ ist

a) eine Tree Edge oder eine Forward Edge genau dann, wenn $d[u] < d[v] < f[v] < f[u]$,

b) eine Back Edge genau dann, wenn $d[v] < d[u] < f[u] < f[v]$, und

c) eine Cross Edge genau dann, wenn $d[v] < f[v] < d[u] < f[u]$.

Aufgabe 3. Betrachten Sie einen Graphen in Adjazenzmatrixdarstellung.

a) Welche Laufzeit hat die Breitensuche auf diesem Graphen?

b) Welche Laufzeit hat die Tiefensuche auf diesem Graphen?

Aufgabe 4. Wahr oder falsch? Falls es in einem gerichteten Graphen einen Pfad von u nach v gibt, dann liefert jede Tiefensuche in dem Graphen das Ergebnis $d[v] < f[u]$.

Aufgabe 5. Die Transponierte eines gerichteten Graphen $G = (V, E)$ ist der Graph $G^T = (V, E^T)$, wobei

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

G^T entsteht also aus G, indem man die Richtung der Kanten umkehrt.

a) Gegeben Sie effiziente Algorithmen an, die die Transponierte eines Graphen berechnen, der als Ajdazenzliste bzw. Adjazenzmatrix vorliegt.

b) Schätzen Sie die Laufzeit Ihrer Algorithmen ab.

Aufgabe 6. Sei $G = (V, E)$ ein zusammenhängender ungerichteter Graph. Entwickeln Sie einen Algorithmus, der einen Pfad durch G berechnet auf dem jede Kante in jede Richtung genau einmal benutzt wird. Welche Laufzeit hat Ihr Algorithmus?