

Algorithmen und Datenstrukturen

Aufgabe 1 Repräsentation von Graphen (Beispiellösung)

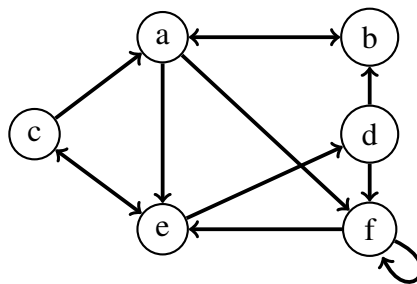
a) Adjazenz-Matrix:

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
<i>a</i>	0	1	0	1	0
<i>b</i>	1	0	1	0	1
<i>c</i>	0	1	0	0	1
<i>d</i>	1	0	0	0	1
<i>e</i>	0	1	1	1	0

Adjazenz-Liste:

a → *b* → *d*
 ↓
b → *a* → *c* → *e*
 ↓
c → *b* → *e*
 ↓
d → *a* → *e*
 ↓
e → *b* → *c* → *d*

b) Gerichteter (!) Graph:



Adjazenz-Liste:

a → *b* → *e* → *f*
 ↓
b → *a*
 ↓
c → *a* → *e*
 ↓
d → *b* → *f*
 ↓
e → *c* → *d*
 ↓
f → *e* → *f*

Aufgabe 2 Baum-Traversierung (Beispiellösung)

- a) Pre-Order (1, 2, 4, 7, 5, 8, 3, 6, 9)
In-Order (7, 4, 2, 8, 5, 1, 6, 9, 3)
Post-Order (7, 4, 8, 5, 2, 9, 6, 3, 1)
- b) Pre-Order (1, 2, 4, 5, 8, 9, 3, 6, 7)
In-Order (4, 2, 8, 5, 9, 1, 6, 3, 7)
Post-Order (4, 8, 9, 5, 2, 6, 7, 3, 1)

Aufgabe 3 Verifikation (Beispiellösung)

- a) Vor- und Nachbedingung von $\mathbf{max}(X)$:

$$C_0 = (n > 0)$$
$$C_4 = (maxval = f(n))$$

- b) Schleifeninvariante und Vorbedingung für die Schleife $\mathbf{while} (i < n) \{ \dots \}$:

$$P = (maxval = f(i))$$
$$C_1 = C_0 \wedge (maxval = X[0]) \wedge (i = 1)$$

- c) Zu zeigen sind:

- Bedingung 1: Schleifeneintritt: $C_1 \Rightarrow P$
Trivial, es gilt $i = 1$ und damit $f(1) = X[0] = maxval$.
- Bedingung 2: Schleifenkörper: $\{C_2 = P \wedge (i < n)\} \dots \{C_3 = P\}$
Wir schreiben die neuen Werte mit Hilfsvariablen:

$$maxval' = \max(maxval, X[i])$$
$$i' = i + 1$$

Soll bei C_3 die Invariante P gelten, so betrachten wir

$$f(i') = f(i + 1)$$
$$= \max(f(i), X[i])$$
$$= \max(maxval, X[i])$$
$$= maxval'$$

Dabei gilt die letzte Zeile, da zuvor die Invariante P ja erfüllt war, und insgesamt gilt P auch am Ende des Schleifenkörpers.

- Bedingung 3: Schleifenende: $(P \wedge \neg(i < n)) \Rightarrow C_4$
Wir betrachten den ersten Fall mit $\neg(i < n)$, also $i = n$. Dann gilt:

$$P \wedge (i = n) \Rightarrow f(i) = f(n) = maxval$$

Dies ist genau die Nachbedingung C_4 !