Prüfung 2009 II
Datenstrukturen & Algorithmen

Universität Bern

Frühling 2018

Prüfung 2009

- Aufgabe 5
- Aufgabe 8
- Aufgabe 13

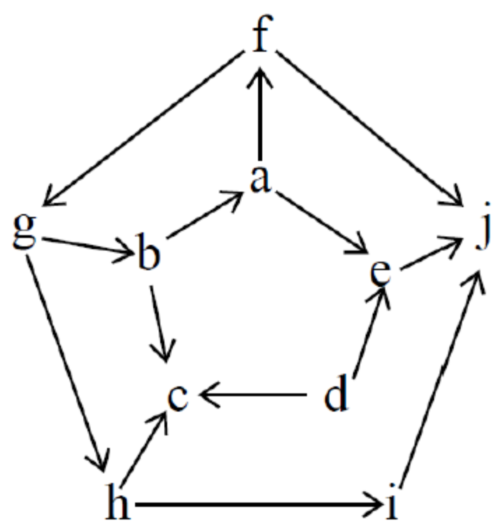
Prüfung 2009

> Aufgabe 5

Der folgende gerichtete Graph wird mit Breitensuche traversiert. Die Suche startet beim Knoten a.

a) Geben Sie eine Reihenfolge an, in der die Knoten erreicht werden können. **1 Punkt**

b) Skizzieren Sie den entsprechenden Breitensuchbaum. **1 Punkt**

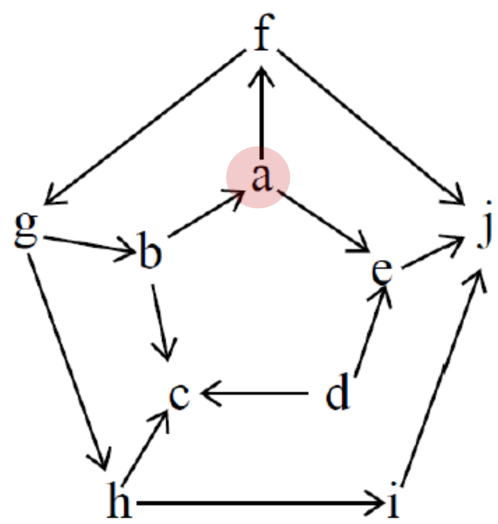


Prüfung 2009

> Aufgabe 5a)

Der folgende gerichtete Graph wird mit Breitensuche traversiert. Die Suche startet beim Knoten a.

a) Geben Sie eine Reihenfolge an, in der die Knoten erreicht werden können.
1 Punkt

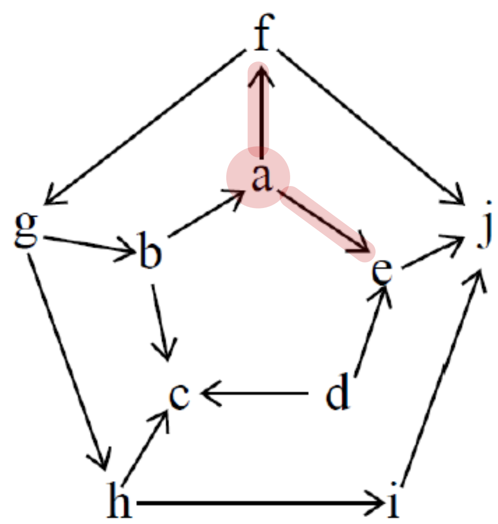


Prüfung 2009

> Aufgabe 5a)

Der folgende gerichtete Graph wird mit Breitensuche traversiert. Die Suche startet beim Knoten a.

a) Geben Sie eine Reihenfolge an, in der die Knoten erreicht werden können.
1 Punkt

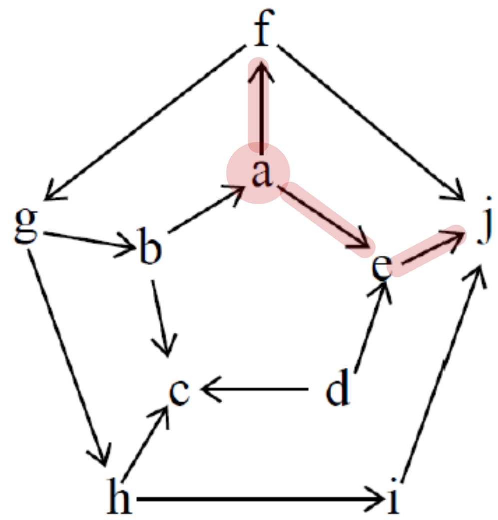


Prüfung 2009

> Aufgabe 5a)

Der folgende gerichtete Graph wird mit Breitensuche traversiert. Die Suche startet beim Knoten a.

a) Geben Sie eine Reihenfolge an, in der die Knoten erreicht werden können.
1 Punkt

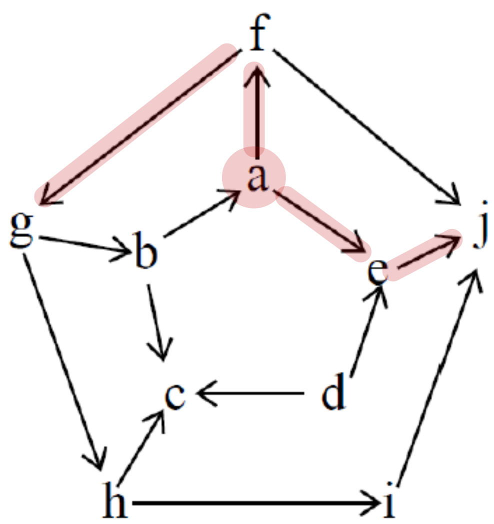


Prüfung 2009

> Aufgabe 5a)

Der folgende gerichtete Graph wird mit Breitensuche traversiert. Die Suche startet beim Knoten a.

a) Geben Sie eine Reihenfolge an, in der die Knoten erreicht werden können.
1 Punkt

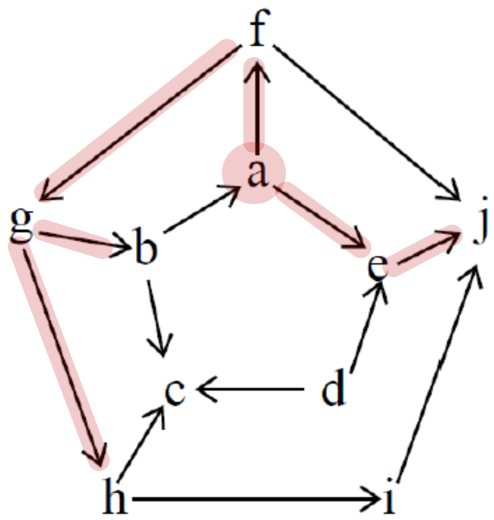


Prüfung 2009

> Aufgabe 5a)

Der folgende gerichtete Graph wird mit Breitensuche traversiert. Die Suche startet beim Knoten a.

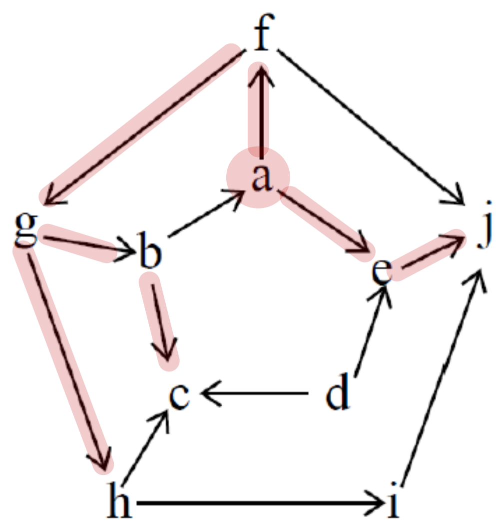
a) Geben Sie eine Reihenfolge an, in der die Knoten erreicht werden können.
1 Punkt



> Aufgabe 5a)

Der folgende gerichtete Graph wird mit Breitensuche traversiert. Die Suche startet beim Knoten a.

a) Geben Sie eine Reihenfolge an, in der die Knoten erreicht werden können.
1 Punkt

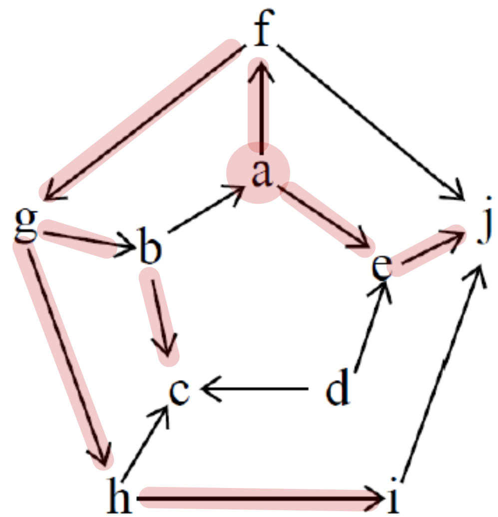


Prüfung 2009

> ## Aufgabe 5a)

Der folgende gerichtete Graph wird mit Breitensuche traversiert. Die Suche startet beim Knoten a.

a) Geben Sie eine Reihenfolge an, in der die Knoten erreicht werden können.
1 Punkt



> → a, e, f, j, g, b, h, c, i

Prüfung 2009

UNIVERSITÄT BERN

> **Aufgabe 5b)**

Der folgende gerichtete Graph wird mit Breitensuche traversiert. Die Suche startet beim Knoten a.

b) Skizzieren Sie den entsprechenden Breitensuchbaum. **1 Punkt**

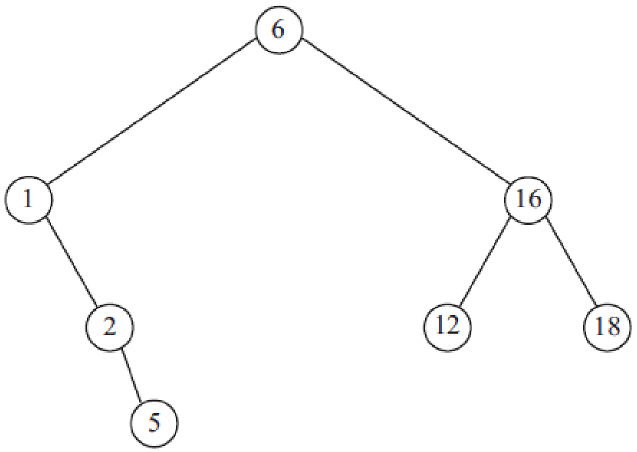
> Nicht eindeutig, abhängig von Lösung in a)
> → Buch Kapitel 22 (Abbildung 22.3)

Prüfung 2009

UNIVERSITÄT BERN

> Aufgabe 8

Gegeben ist der binäre Suchbaum in der Figur unten.

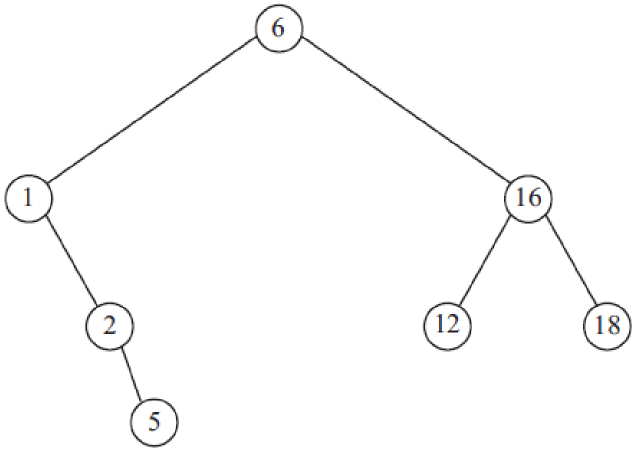


a) Geben Sie die Reihenfolge der Knoten in der Inorder und der Preorder-Traversierung. **2 Punkte**

c) Zeichnen Sie den binären Suchbaum, dessen Postorder-Traversierung die Folge $1, 4, 5, 3, 2, 6, 9, 11, 10, 8, 7, 13, 12$ ergibt. Hinweis: Überlegen Sie sich zuerst, welches die Wurzel des Baumes ist. **2 Punkte**

Prüfung 2009

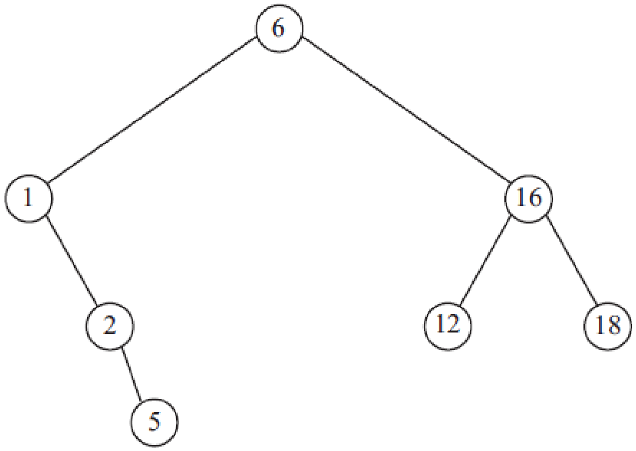
> **Aufgabe 8a) Inorder**
  - Traversiere linken Teilbaum
  - Besuche Wurzel
  - Traversiere rechten Teilbaum
> → Alle Schlüssel in sortierter Reihenfolge ausgeben



> → 1, 2, 5, 6, 12, 16, 18

Prüfung 2009

> **Aufgabe 8a) Preorder**
- Besuche Wurzel
- Traversiere linken Teilbaum
- Traversiere rechten Teilbaum

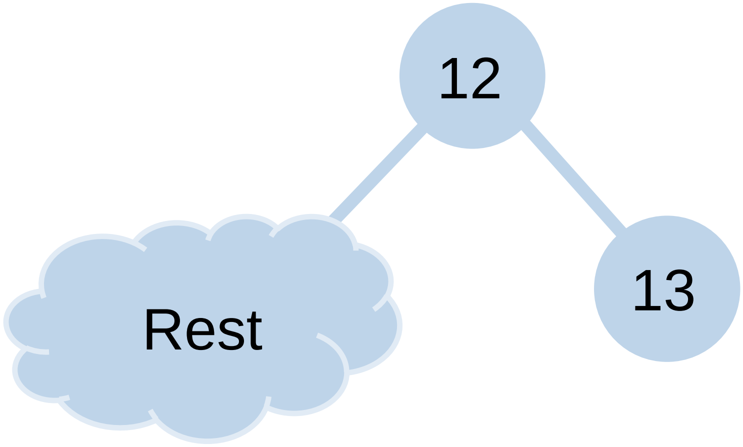


> 6, 1, 2, 5, 16, 12, 18

Prüfung 2009

> **Aufgabe 8c)**

c) Zeichnen Sie den binären Suchbaum, dessen Postorder-Traversierung die Folge 1, 4, 5, 3, 2, 6, 9, 11, 10, 8, 7, 13, 12 ergibt. Hinweis: Überlegen Sie sich zuerst, welches die Wurzel des Baumes ist. **2 Punkte**



> **Postorder**

— Traversiere linken Teilbaum

— Traversiere rechten Teilbaum

— Besuche Wurzel

> → Die Wurzel ist die 12

> **Binärer Suchbaum**

— Schlüssel in linkem Teilbaum ≤ Knoten

— Schüssel in rechtem Teilbaum ≥ Knoten

Prüfung 2009

> **Aufgabe 13** Taxifahrt durch gefährliches Quartier

- **a)** Rekursives Programm für Gesamtrisiko, Laufzeit **2 Punkte**
- **b)** Programme mit dynamischer Programmierung für Gesamtrisiko, Laufzeit **3 Punkte**
- **c)** Beschreibe Rückverfolgung in der Lösungstabelle von b) um sichersten Pfad zu finden, Laufzeit **2 Punkte**
- **d)** Dynamische Programmierung erfolgreich, wenn keine Einbahnstrassen? **1 Punkt**

	9	1	2	3	5	7
8	3	3	11	2	9	8
12	9	4	9	2	14	1
7	10	8	16	3	5	9
1	2	2	12	15	8	

Prüfung 2009

> **Aufgabe 13a)** Erstelle ein **rekursives** Programm in Pseudocode, welches das Gesamtrisiko eines sichersten Weges berechnet. Gib die Laufzeit des Algorithmus an.

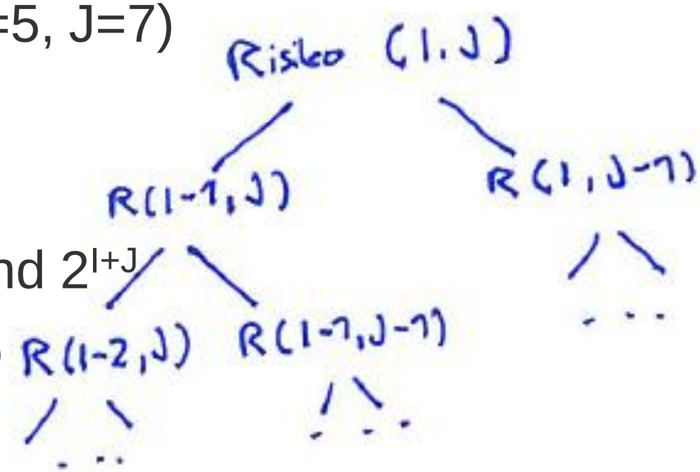
```
// risk(i, j): Gesamtrisiko des Pfades r(1,1) -> r(i,j)
risk(i, j)
    if (i = 1 && j = 1) return 0;      // Abbruch
    if (i < 1 || j < 1) return inf;    // oops!
    return r[i, j]
        + min(risk(i-1, j), risk(i, j-1))
```

> → Kosten für sichersten Weg: `risk(5, 7)`

Prüfung 2009

UNIVERSITÄT BERN

- **Aufgabe 13a)** Laufzeit des Algorithmus
- Zeichne Rekursionsbaum
  - I, J ist Grösse des Strassennetzes (I=5, J=7)
- Betrachte Höhe des Baumes `h`
  - **Es gilt** $\min(I, J) < h < I + J$
  - → Anzahl Knoten zwischen $2^{\min(I, J)}$ und 2^{I+J}
- → Laufzeit $O(2^{I+J})$ und $\Omega(2^{\min(I, J)})$




```
// risk(i, j): Gesamtrisiko des Pfades r(1,1) -> r(i,j)
risk(i, j)
    if (i = 1 && j = 1) return 0;       // Abbruch
    if (i < 1 || j < 1) return inf;    // oops!
    return r[i, j]
        + min(risk(i-1, j), risk(i, j-1))
```

Prüfung 2009

> **Aufgabe 13b)** Entwirf einen Algorithmus nach dem Muster der **dynamischen Programmierung**, der das Gesamtrisiko eines sichersten Weges berechnet. Gib die Laufzeit des Algorithmus an.

	9	1	2	3	5	7
8	3	3	11	2	9	8
12	9	4	9	2	14	1
7	10	8	16	3	5	9
1	2	2	12	15	8	

```
I=5, J=7
c[1,1] = 0
for (i=1:I)
    for (j=1:J)
        if (i != 1 || j!=1)
            // Konvention: c[i,j] = inf für i<1 || j<1
            c[i,j] = min(c[i-1,j], c[i,j-1]) + r[i,j]

return c[I,J] // Gesamtrisiko
```

Prüfung 2009

UNIVERSITÄT BERN

> **Aufgabe 13b)** Entwirf einen Algorithmus nach dem Muster der **dynamischen Programmierung**, der das Gesamtrisiko eines sichersten Weges berechnet. Gib die Laufzeit des Algorithmus an.

```
I=5, J=7
c[1,1] = 0
for (i=1:I)
    for (j=1:J)
        if (i != 1 || j!=1)
            // Konvention: c[i,j] = inf für i<1 || j<1
            c[i,j] = min(c[i-1,j], c[i,j-1]) + r[i,j]

return c[I,J] // Gesamtrisiko
```

Laufzeit Θ`(i*j)`

Prüfung 2009

UNIVERSITÄT BERN

- **Aufgabe 13c)** Beschreibe in Worten, wie durch **Rückverfolgung** in der Lösungstabelle, die dein Algorithmus aus b) berechnet, die traversierten Zellen eines sichersten Weges gefunden werden können. Gib die **Laufzeit** für die Rückverfolgung an.

- → Konstruiere den Weg Feld für Feld rückwärts, beginnend bei `c[I,J]`. Das vorhergehende Wegfeld ist jeweils `min(c[i-1,j], c[i,j-1])`. Sind beide Felder gleich teuer, führen beide Wahlen zu einem optimalen Weg.

Prüfung 2009

> **Aufgabe 13c)** Beschreibe in Worten, wie durch **Rückverfolgung** in der Lösungstabelle, die dein Algorithmus aus b) berechnet, die traversierten Zellen eines sichersten Weges gefunden werden können. Gib die **Laufzeit** für die Rückverfolgung an.

> → Konstruiere den Weg Feld für Feld rückwärts, beginnend bei `c[I,J]`. Das vorhergehende Wegfeld ist jeweils `min(c[i-1,j], c[i,j-1])`. Sind beide Felder gleich teuer, führen beide Wahlen zu einem optimalen Weg.

Laufzeit $\Theta(i+j)$

Prüfung 2009

UNIVERSITÄT BERN

> **Aufgabe 13d)** Kann das Problem immer noch mittels dynamischer Programmierung gelöst werden, wenn die Strassen in beide Richtungen befahrbar sind? Begründe deine Antwort.

> → Betrachte folgendes Problem

```
c[i,j] = r[i,j] + min(c[i-1,j], c[i,j-1],
                      c[i+1,j], c[i,j+1])
```

> → Hat das Problem eine optimale Teilstruktur?

— Kann es auf kleinere Teilprobleme zurückgeführt werden?