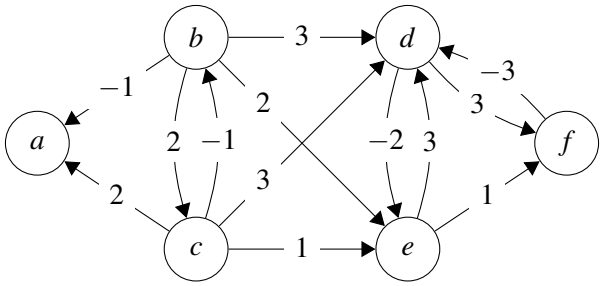
Aufgabe 1. Kleinaufgaben [20 Punkte]

a. Enthält folgender gerichteter gewichteter Graph einen kürzesten Weg von b nach f?



Falls ja, geben Sie einen solchen kürzesten Weg an. Falls nein, begründen Sie kurz, warum kein solcher kürzester Weg existiert. [2 Punkte]

b. Betrachten Sie noch einmal den gerichteten gewichteten Graph aus Teilaufgabe **a.** Enthält er einen kürzesten Weg von c nach a? Falls ja, geben Sie einen solchen kürzesten Weg an. Falls nein, begründen Sie kurz, warum kein solcher kürzester Weg existiert. [1 Punkt]

c. Sei A eine $(n \times n)$-Matrix über $\mathbb{R}$ und $n, m \in \mathbb{N}_{>0}$. Kann A^m mit höchstens $O(\log m)$ Matrixmultiplikationen berechnet werden? Begründen Sie kurz. [2 Punkte]

(weitere Teilaufgaben auf den nächsten Blättern)

Fortsetzung von Aufgabe 1

d. Zeigen Sie oder widerlegen Sie, dass $5^{\log_3 n} = \mathrm{O}\left(n^2\right)$ gilt. [2 Punkte]

e. Zeigen Sie oder widerlegen Sie, dass $2^n = \Omega(2^{n/2})$ gilt. [2 Punkte]

f. Gegeben seien zwei rekursive Algorithmen A und B, wobei B in jedem Rekursionsschritt nicht nur sich selbst sondern auch A aufruft. Die Laufzeiten von A und B seien durch die Rekurrenzen

$$\begin{aligned} T_A(n) &= 5n + aT_A(n/3), \qquad T_A(1) = 4 \quad \text{bzw.} \\ T_B(n) &= T_A(n) + bT_B(n/3), \quad T_B(1) = 2 \end{aligned}$$

beschrieben mit $n = 3^k$ und $k \in \mathbb{N}_{>0}$. Geben Sie Werte für a und b aus $\mathbb{N}_{>0}$ an, so dass die Laufzeit von B in $\Theta(n \log n)$ liegt für $n = 3^k$. Begründen Sie Ihre Wahl von a und b kurz. [3 Punkte]

g. Geben Sie in Θ-Notation an, welches Worst-Case-Laufzeitverhalten ein vergleichsbasierter Sortieralgorithmus bestenfalls haben kann. Nennen Sie ein Beispiel für einen vergleichsbasierten Sortieralgorithmus, der in diesem Sinne optimal und zusätzlich stabil ist. [2 Punkte]

(weitere Teilaufgaben auf dem nächsten Blatt)

Fortsetzung von Aufgabe 1

h. Gegeben seien n Bausteine mit Höhen $h_1, \ldots, h_n \in \mathbb{N}_{>0}$. Die Bausteine sollen auf zwei Stapel verteilt werden. Ziel ist es, die Gesamthöhe des höheren Stapels zu minimieren.

Dieses Problem soll optimal gelöst werden. Dazu schlägt ein Student folgenden Ansatz vor:

> Berechne die Lösung für die ersten $k+1$ Steine rekursiv:
>
> - Berechne die Lösung für die ersten k Steine. Falls $k = 0$ starte mit zwei leeren Stapeln.
> - Platziere den $(k+1)$-ten Stein auf dem kleineren Stapel, oder auf dem ersten Stapel falls beide gleich hoch sind.

Liefert dieser Ansatz immer eine optimale Lösung? Wenn ja, begründen Sie kurz. Wenn nein, geben Sie ein Gegenbeispiel an. [3 Punkte]

i. Gegeben sei die sortierte Sequenz $1, 5, 6, 8, 11, 12, 13$. Geben Sie für diese Sequenz einen zulässigen $(2,4)$-Baum an, der eine maximale Anzahl innere Knoten enthält. Vervollständigen Sie dazu die unten angegebene Zeichnung – die Blätter des $(2,4)$-Baumes sind bereits eingezeichnet. [3 Punkte]

Aufgabe 2. Graphrepräsentationen [7 Punkte]

Gegeben sei ein Graph $G = (V, E)$ mit $V := \{1, 2, \ldots, 8\}$. Die Kanten seien durch die Adjazenz-Matrix $A \in \{0,1\}^{8\times 8}$ beschrieben:

$$A := \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

a. Geben Sie den Graphen dargestellt als Adjazenzfeld an. Benutzen Sie dafür die vorgegebene Zeichnung. [3 Punkte]

Tragen Sie **hier** die **Lösung** ein:

	1	2	3	4	5	6	7	8	9
V									

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
E															

Kopie der obigen Zeichnung zum **Rechnen:**

	1	2	3	4	5	6	7	8	9
V									

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
E															

(Teilaufgaben **b.** und **c.** auf dem nächsten Blatt)

Fortsetzung von Aufgabe 2

Kopie der Adjazenz-Matrix vom **vorherigen** Blatt:

$$A := \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

b. Zeichnen Sie den Graphen. [2 Punkte]

c. Handelt es sich bei dem Graphen um einen DAG? Begründen Sie kurz. [2 Punkte]

Aufgabe 3. Verkettete Listen [8 Punkte]

Betrachten Sie eine Realisierung zyklischer einfach verketteter Listen, bei der die Listenglieder und die Liste durch die folgenden zwei Klassen beschrieben sind:

```
1: class ListItem of Element
2:     e : Element
3:     next : Handle
4: end class
```

```
1: class List of Element
2:     head := (⊥, address of head) : ListItem of Element
3: end class
```

Die leere Liste wird dabei dargestellt, indem das Dummyglied *head* auf sich selbst verweist, also $head.next = head$.

Ziel dieser Aufgabe ist es einen **iterativen** Algorithmus anzugeben, der in $O(n)$ Zeit aus einer gegebenen Liste $\langle e_1, e_2, \ldots, e_n \rangle$ die Liste $\langle e_n, e_{n-1}, \ldots, e_1 \rangle$ erzeugt, die Liste also umdreht. Es soll aber **keine** neue Liste erzeugt werden. Vielmehr soll die Eingabeliste selbst umgedreht werden.

a. Geben Sie zunächst eine aussagekräftige Schleifeninvariante an, aus der die Korrektheit Ihres Algorithmus folgt. [3 Punkte]

b. Geben Sie nun – basierend auf Ihrer Invariante aus Teilaufgabe **a.** – Pseudocode für eine Methode *umdrehen* der Klasse *List* an.[1] [5 Punkte]

method *umdrehen*

[1] Grundlage für die Bewertung sind nicht syntaktische Details sondern der iterative Algorithmus.

Aufgabe 4. Urlaubsdatenbank [9 Punkte]

In einem Unternehmen habe jeder Mitarbeiter eine eindeutige ID aus $\mathbb{N}$. Gegeben sei eine Datei D (d.h. eine Folge) von n Paaren der Form $(\mathit{MitarbeiterID}, \mathit{UrlaubsTag}) \in \mathbb{N} \times \{1, \ldots, 365\}$. Die Datei D beschreibt für ein bestimmtes Jahr, an welchen Tagen die Mitarbeiter Urlaub genommen haben: Für jeden Tag, an dem ein Mitarbeiter in dem Jahr Urlaub genommen hat, enthält D genau ein entsprechendes Paar.

Hinweis: Beachten Sie, dass die IDs der Mitarbeiter beliebig groß sein können.

a. Geben Sie einen Algorithmus an, der in **erwartet** $O(n)$ Zeit die IDs aller Mitarbeiter in D ausgibt, die bereits in der ersten Hälfte des Jahres mindestens 30 Tage Urlaub genommen haben. Jede ID darf dabei nicht mehr als einmal ausgegeben werden. [4 Punkte]

b. Begründen Sie kurz, warum Ihr Algorithmus aus Teilaufgabe **a.** das gewünschte Laufzeitverhalten aufweist. [2 Punkte]

(Teilaufgaben **c.** und **d.** auf dem nächsten Blatt)

Fortsetzung von Aufgabe 4

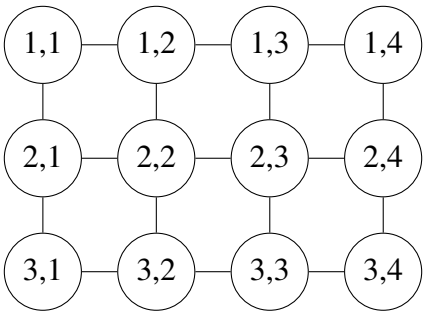
c. Geben Sie einen Algorithmus an, der an Hand von D in **deterministisch** $O(n)$ Zeit einen Tag des Jahres ermittelt, an dem eine maximale Anzahl Mitarbeiter Urlaub genommen hat. [2 Punkte]

d. Begründen Sie kurz, warum Ihr Algorithmus aus Teilaufgabe **c.** das gewünschte Laufzeitverhalten aufweist. [1 Punkte]

Aufgabe 5. Kürzeste Wege [7 Punkte]

a. Gegeben sei ein regelmäßiges $(a \times b)$-Gitter, bei dem zwischen direkt nebeneinander und direkt übereinander liegenden Knoten je eine horizontale bzw. vertikale Kante existiert. Alle diese Kanten sind ungerichtet und haben das Gewicht 1.

Beispiel: 3×4 Gitter:



Der Knoten an i-ter Stelle von oben und j-ter Stelle von links heißt (i, j) wie im Beispiel.

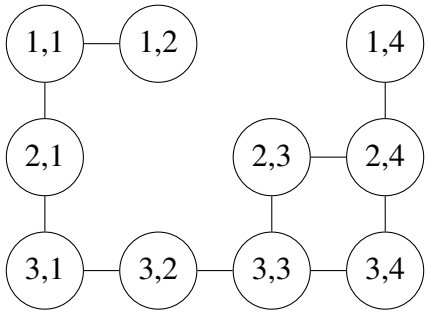
Geben Sie einen Algorithmus an, der einen kürzesten Weg von einem Knoten (i_1, j_1) zu einem Knoten (i_2, j_2) berechnet und ausgibt. Ihr Algorithmus soll eine Laufzeit in O(Weglänge) besitzen. [3 Punkte]

(Teilaufgabe **b.** auf dem nächsten Blatt)

Fortsetzung von Aufgabe 5

b. Gegeben sei ein regelmäßiges $(a \times b)$-Gitter wie in Teilaufgabe **a.**, bei dem nun aber einzelne Knoten fehlen.

Beispiel: 3×4 Gitter mit fehlenden Knoten $(2,2)$ und $(1,3)$:



Beschreiben Sie einen Algorithmus, der einen kürzesten Weg von einem existierenden Knoten (i_1, j_1) zu einem existierenden Knoten (i_2, j_2) berechnet und ausgibt. Falls kein Weg zwischen diesen Knoten existiert soll der Algorithmus "nicht verbunden" ausgeben.

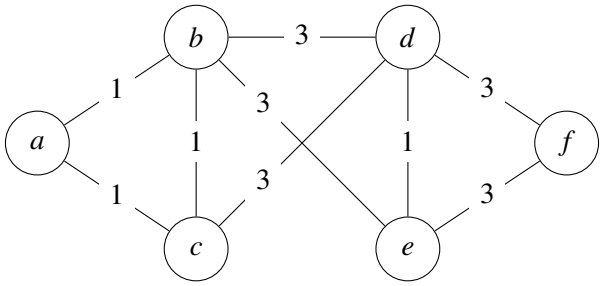
Das Gitter sei in Form eines Adjazenzfeldes gegeben. Ihr Algorithmus soll eine Laufzeit in $O(n)$ besitzen (n := Anzahl existierender Knoten). [4 Punkte]

Aufgabe 6. Minimale Spannbäume [9 Punkte]

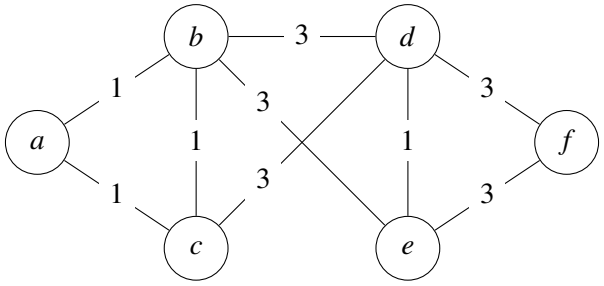
a. Gegeben sei der abgebildete ungerichtete Graph mit Kantengewichten aus $\{1,3\}$. Auf diesem Graph soll der Jarnik-Prim Algorithmus beginnend beim Knoten c ausgeführt werden.

Markieren Sie im Graph die Kanten eines möglichen resultierenden MST. Nummerieren Sie die markierten Kanten außerdem in einer möglichen Reihenfolge, in der diese Kanten vom Jarnik-Prim Algorithmus in den MST aufgenommen werden. [2 Punkte]

Tragen Sie die **Lösung** bitte in **diesen** Graphen ein:



Kopie des obigen Graphen zum **Rechnen:**



(Teilaufgaben **b.** und **c.** auf dem nächsten Blatt)

Fortsetzung von Aufgabe 6

b. Wir betrachten nun beliebige zusammenhängende ungerichtete Graphen $G = (V,E)$ mit $V = \{1,\ldots,n\}$ und Kantengewichten aus $\{1,3\}$. Sei G in Form eines **Adjazenzfeldes** gegeben. Geben Sie einen Algorithmus an, der in $O(|E|)$ Zeit einen MST von G berechnet. [5 Punkte]

c. Argumentieren Sie kurz, warum Ihr Algorithmus aus Teilaufgabe **b.** das gewünschte Laufzeitverhalten aufweist. [2 Punkte]

Konzeptpapier (Abgabe freiwillig)