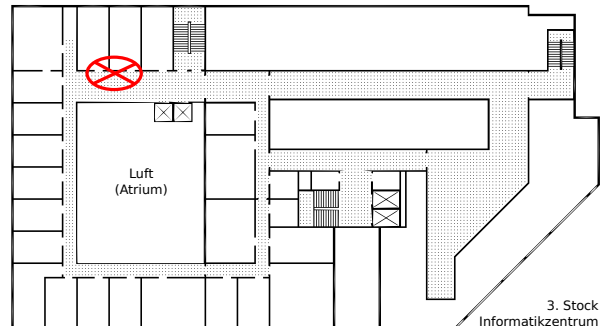


Hausaufgabenblatt 5

Abgabe der Lösungen bis zum 10.07.2026 um 13:15 Uhr im Hausaufgabenschrank bei Raum IZ 337 (siehe Skizze rechts). Es werden nur mit einem dokumentenechten Stift (kein Rot!) geschriebene Lösungen gewertet.

Bitte Namen, Matrikelnummer und Gruppennummer auf die Abgabe schreiben. Mehrere Blätter tackern!



Hausaufgabe 1 (Graphenprobleme):

(2+2+4 Punkte)

In dieser Aufgabe betrachten wir die folgenden zwei Entscheidungsprobleme:

1. INDEPENDENT SET:

Gegeben: Ein Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}$

Frage: Existiert eine Menge $IS \subseteq V$ mit $|IS| \geq k$, sodass für jede Kante höchstens ein Knoten in IS liegt, d.h., $\forall \{u, v\} \in E : |\{u, v\} \cap IS| \leq 1$?

2. VERTEX COVER:

Gegeben: Ein Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}$

Frage: Existiert eine Menge $VC \subseteq V$ mit $|VC| \leq k$, sodass für jede Kante mindestens ein Knoten in VC liegt, d.h., $\forall \{u, v\} \in E : |\{u, v\} \cap VC| \geq 1$?

- Betrachte den Graphen G aus Abbildung 1. Gib ein Independent Set der Größe $k = 2$ an. Gibt es ein größeres Independent Set? Begründe deine Antwort.
- Betrachte weiterhin den Graphen G aus Abbildung 1. Gib ein Vertex Cover der Größe $k = 3$ an. Gibt es ein kleineres Vertex Cover? Begründe deine Antwort.
- Sei $H = (V, E)$ ein Graph.

Zeige: VC ist ein Vertex Cover in H , genau dann wenn $IS = V \setminus VC$ ein Independent Set in H ist.

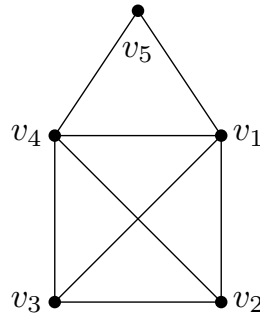


Abbildung 1: Der Graph G .

Hausaufgabe 2 (Integer Programming):

(3 Punkte)

Man kann das Finden eines kleinstmöglichen Vertex Cover in einem Graph $G = (V, E)$ wie folgt als Integer Program (IP) formulieren:

$$\begin{aligned} \min \quad & \sum_{v \in V} x_v \\ \text{subject to:} \quad & x_u + x_w \geq 1 \quad \forall \{u, w\} \in E \\ & x_v \in \{0, 1\} \quad \forall v \in V \end{aligned}$$

Analog kann auch das Finden des größtmöglichen Independent Set als IP formuliert werden. Gib die entsprechende Formulierung an.

Hausaufgabe 3 (Hashing):

(5+4 Punkte)

- a) Betrachte ein Array A der Größe $m = 11$ mit Speicherzellen $A[0], \dots, A[10]$. Das Array enthält zu Beginn keine Elemente. In diesem Array führen wir Hashing mit offener Adressierung mit der folgenden Hash-/Sondierungsfunktion durch:

$$h(x, i) = (h_1(x) + i \cdot h_2(x)) \bmod m, \text{ mit } h_1(x) = 2 \cdot (x + 5)$$

$$\text{und } h_2(x) = (x^2 + 1) \bmod m.$$

Dabei ist x ein Schlüssel und i die Nummer des Versuchs, x in eine unbesetzte Speicherzelle des Arrays einzufügen, beginnend bei $i = 0$. Berechne zu jedem der folgenden Schlüssel die Position, die er in A bekommt:

16, 5, 7, 8, 10

Dabei sollen die Schlüssel in der gegebenen Reihenfolge eingefügt werden und der Rechenweg soll klar erkennbar sein. Gib insbesondere alle berechneten Werte von $h(x, i)$ mit Rechenweg an. Gib außerdem den Endzustand des Arrays in einer Tabelle mit dem folgenden Format an.

j	0	1	2	3	4	5	6	7	8	9	10
$A[j]$											

- b) Nimm an, wir nutzen Hashing mit offener Adressierung und Löschoperationen mittels der in der Vorlesung vorgestellten *lazy deletions*, also mit den Markierungen der Felder als *besetzt*, *noch nie besetzt*, oder *wieder frei*.

Beschreibe kurz (in Fließtext oder Pseudocode), wie ein Element in einer Hashtabelle mit offener Adressierung gesucht wird. Gehe insbesondere darauf ein, wann man keine weiteren Versuche mehr betrachten muss. Du kannst annehmen, dass in der Hashtabelle lineares Sondieren genutzt wird.