

8. Informatik-Olympiade vom Land Brandenburg

– Vorrunde –

Hinweise:

Die Aufgaben werden von einzelnen Personen bearbeitet und unter Erbringung der eigenen Leistung. Die Abgabe erfolgt (inklusive der kompletten Namens- und Schul-Nennung) an informatik@blis-brandenburg.de. Häufig kommt es zu Problemen bei dem Empfang von exe-Dateien. In diesem Fall bitte zunächst versuchen, die exe-Datei einfach als txt abzuspeichern mit dem Hinweis, dass es sich um eine exe-Datei handelt. Im Notfall geht auch ein Link zu einem Upload-Ordner von euch, ansonsten schicken wir auf Anfrage einen Link zu einem Upload-Ordner.

Der Abgabe-Ordner soll einen Programm-Ordner mit den kompletten Dateien enthalten, sodass das Programm ausführbar ist, sowie eine erklärende Dokumentation zum Programm (Aufgabe 2) und das Dokument zur Lösung der theoretischen Aufgabe (Aufgabe 1). Bei den Dokumenten sollen möglichst auch visuelle Darstellungen die Erklärung unterstützen.

Auswirkungen der Vorrunde auf den Hauptwettbewerb (Informatik-Olympiade am 05./06.06.2025):

Die beiden Aufgabentypen bereiten euch als Teilnehmer*innen gleichzeitig auf die beiden Olympiade-Tage vor.

- Die Teilnahme an der Vorrunde erzeugt ein Ranking für die Teilnehmerbewerbung am Hauptwettbewerb, sodass eine vorzeitige Bekanntgabe der persönlichen Teilnahme bei der Olympiade ermöglicht werden kann.
- Die Teilnahme an der Vorrunde ist auch in diesem Jahr weiterhin noch KEINE Voraussetzung zur Teilnahme am Hauptwettbewerb im Juni, allerdings werden die Teilnehmer aus der Vorrunde bevorzugt zur Hauptrunde zugelassen.

Alle Teilnehmer*innen der Vorrunde starten, je nach Platzierungen (Platz 1-3 und dann der Rest), mit unterschiedlichem Punktevorsprung in den Hauptwettbewerb.

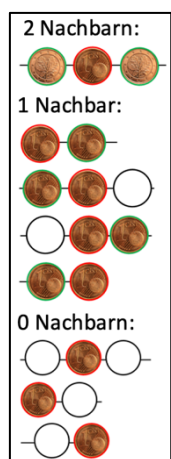
Aufgabe 1: Münzspiel

Es handelt sich um eine Art Spiel für eine Einzelperson, was man auch sehr gerne einfach mal nachspielen kann.

Angenommen wir haben 10-12 Münzen, deren eine Seite Kopf zeigt und die andere Zahl ist.

Die Münzen werden in einer Linie zufällig angeordnet.

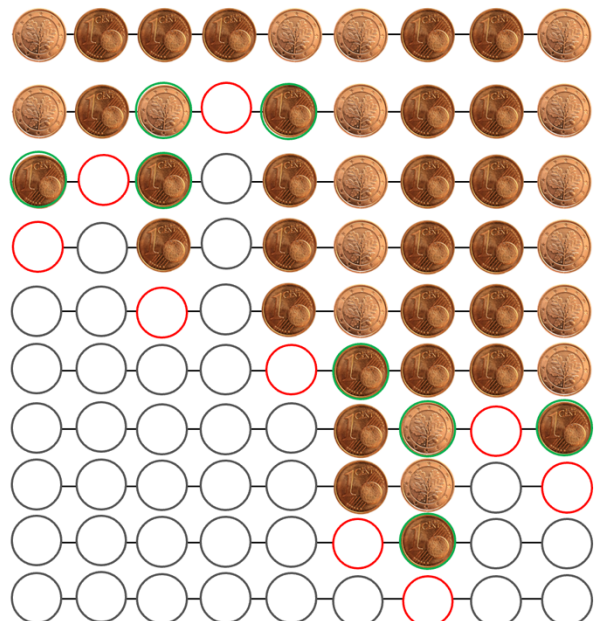
Dann kann das Spiel beginnen. Pro Spielzug wird eine Zahl-Münze ausgewählt und entfernt während die direkten Nachbar-Münzen umgedreht werden und somit ihre Deckseite wechseln. Wenn eine Münze entfernt wird, bleibt diese Stelle frei und gilt auch nicht mehr als Nachbar.



Die Definition als Nachbar ist immer auf die ursprüngliche Aneinanderreihung bezogen. Das bedeutet, durch das Entfernen von Münzen entstehen Lücken, die dazu führen, dass eine Münze nicht immer seine zwei Nachbarn hat, sondern auch mal einen oder keinen Nachbarn hat (siehe Abb. links).

Ziel ist es, alle Münzen durch schlaue Spielzüge zu entfernen.

In der Abb. rechts ist beispielsweise ein Spielablauf dargestellt, bei dem es dem Spieler gelingt alle Plättchen zu entfernen. Hierbei sind die Münzentfernungen rot und die Umdrehung grün markiert.



Aufgabe: Entwickeln Sie eine Strategie (ohne Computerprogramm), um diese Spiele (mit beliebiger Start-Anordnung) zu gewinnen. Nennen Sie Bedingungen für Gegebenheiten, die für einen Gewinn notwendig sind und erklären Sie die Allgemeingültigkeit dieser.

Aufgabe 2: Finde den Ausgang

2.1 Szenario

Ein Informatik-Student befindet sich in einer Art Escape-Room. Ziel des Escape-Rooms ist es, dass der Studierende schnellstmöglich zum Ausgang gelangt. Dabei versperren ihm einige Türen den Weg. Er braucht zum Knacken einer Tür immer eine gewisse Zeit und kann darum immer nur eine bestimmte Anzahl an Türen während des Escape-Games öffnen.

2.2 Aufbau der Eingabe und Ausgabe

Das Programm soll eine $m \times n$ -Matrix erhalten, in dem jede Zelle entweder 0 (leer) oder 1 (Tür) ist. Sie können in einem Schritt von und zu einer leeren Zelle nach oben, unten, links oder rechts wechseln.

Der Studierende startet immer in dem obersten linken Feld $(0,0)$ eines Spielfelds und muss zum Ausgang gelangen, der immer in der untersten rechten Ecke $(m - 1, n - 1)$ befindet. Pro Spieldurchlauf wird immer angegeben, wie viele (k) Türen der Studierende schafft zu öffnen, um noch rechtzeitig den Ausgang zu erreichen.

- $m =$ Höhe der Matrix
- $n =$ Breite der Matrix
- $1 \leq m, n \leq 40$
- $1 \leq k \leq m \cdot n$
- $\text{matrix}[i][j] == 0$ oder 1
- $\text{matrix}[0][0] == \text{matrix}[m - 1][n - 1] == 0$

Ausgegeben werden soll die Mindestanzahl von Schritten, um vom Start zum Ausgang zu gelangen, mit der gegebenen Bedingung, dass der Studierende höchstens k Türen öffnen kann.

Es soll immer die Mindestanzahl von Schritten und der Weg (nacheinander Angabe der Felder) ausgegeben werden, sowie die zu öffnenden Türen ausgegeben werden.

2.3 Beispiele für Grundaufgabe

Beispiel 1:



Eingabe: Matrix = $[[0,0,0],[1,1,0],[0,0,0],[0,1,1],[0,0,0]]$, $k = 1$

Ausgabe: 6

Erklärung: Der kürzeste Weg ohne Türen zu öffnen ist 10. Der kürzeste Weg mit einer Türöffnung an Position $(3,2)$ ist 6. Dieser Pfad ist $(0,0) \rightarrow (0,1) \rightarrow (0,2) \rightarrow (1,2) \rightarrow (2,2) \rightarrow (3,2) \rightarrow (4,2)$.

Beispiel 2:



Eingabe: Matrix = $[[0,1,1],[1,1,1],[1,0,0]]$, $k = 1$

Ausgabe: Nicht möglich

Erklärung: Es müssen mindestens 2 Türen geöffnet werden, um einen Weg zu finden.

2.4 Aufgabe

Schreiben Sie ein Programm, welches folgendes erfüllt und den Informatik-Student zum Ausgang führt.

- Wenn es einen Weg gibt, der ohne Türen zu öffnen geht, dann geben Sie den kürzesten davon an. Geben Sie in diesem Fall aber auch den kürzesten Weg mit Öffnung der Türen an (sofern dieser kürzer ist) unter der Angabe welche Türen dafür geöffnet werden müssen.
- Wenn es nicht möglich ist, einen solchen Weg zu finden, geben Sie dies an und danach eine Möglichkeit, für einen kürzesten Weg mit den nächstmöglich wenigsten Türöffnungen.
- Erzeugen Sie Beispiele-Eingabefelder, bei denen jeweils die Fälle aus a) und b) erfüllt sind.
- Schreiben Sie ein Teilprogramm, das die Arbeit des Escape-Room-Entwicklers übernimmt. Es soll z.B. ein 40×40 Spielfeld erzeugt werden, bei dem der Studierende mindestens 80 Schritte braucht und 10 Türen knacken muss.