

6. 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 oder, sollte das nicht funktionieren, direkt als Upload (innerhalb eines Ordners mit Namen), den Link dafür erhaltet ihr als Antwort auf eine Anfrage-Mail von euch. Der Abgabe-Ordner soll einen Programm-Ordner mit den kompletten Dateien enthalten, sodass das Programm ausführbar ist, sowie eine erklärende Dokumentation zum Programm (Aufgabe 1) 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 02./03.06.2023):

- Die beiden Aufgabentypen bereiten euch als Teilnehmer*innen gleichzeitig auf die beiden Olympiaden-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.

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

Aufgabe 1: Schokolade zerteilen

Otto möchte eine Tafel Schokolade in die einzelnen Stücke zerteilen. Eine Tafel besteht aus $n \cdot m$ einzelnen quadratischen Stücken. Um ein Schokoladenstück zu zerteilen, kann Otto das Stück entlang der Trennlinie aufbrechen.

Beispiel: Sei $n = 9$ und $m = 5$

Im einfachsten Fall bricht Otto die Tafel zunächst in 9 Riegel zu je 5 Stücke und anschließend jeden Riegel durch 4 Teilungen in seine 5 Stücke. Das sind zusammen $8 + 9 \cdot 4 = 44$ Teilungen. Oder aber er teilt die Tafel in ...

Aufgabe:

Gibt die kleinste Zahl von Teilungen an, die Otto benötigt, um eine $n \cdot m$ Tafel Schokolade in ihre Stücke zu zerlegen. Begründe deine Lösung.



Aufgabe 2: Kekse, ich brauche Kekse!

2.1 Szenario

Oh nein! Gleich geht Deine Informatikklausur los und Du wolltest im Supermarkt noch schnell Kekse besorgen.

An der Kasse angekommen musst Du allerdings feststellen, dass sich eine riesige Schlange vor Dir gebildet hat. Wenn Du Dich hier anstellst, wirst Du sicher zu spät kommen! Deine einzige Chance ist also, Dich vorzudrängeln, auch wenn Du sowas normalerweise eigentlich nicht machst.



Um nicht aufzufallen, drängelst Du Dich nur unter folgenden Regeln in der Schlange vor:

Nur dort, wo gerade größere Lücken zwischen zwei Kund:innen sind, kannst Du Dich in die Schlange mogeln. Stehen die Kund:innen zu dicht, würde es auffallen, wenn Du Dich zwischen sie quetschst. Außerdem ist Dein Einkaufskorb so schwer, dass Du nie an mehr als 10 Positionen (an jeder Position ist entweder genau ein:e Kund:in oder genau eine Lücke) auf einmal vorbeilaufen kannst, ohne dass Dir die Puste ausgeht. Zu guter Letzt möchtest Du Dich so selten wie möglich vordrängeln, damit die Chance, erwischt zu werden, gering bleibt.

2.2 Aufbau der Eingabedatei

Du erhältst als Eingabe eine einzige Zeile. Diese enthält mit Leerzeichen getrennt die Nummern all dieser Positionen in der Schlange, an denen eine Lücke ist. Du stehst zu Beginn an Position 0 in der Schlange, die Positionen werden in Richtung der Kasse größer.

2.3 Beispiel

Für die Eingabe: 5 8 12 13 17 25 26 30 35

sähe die Schlange also wie folgt aus: DKKKK_KK_KKK__KKK_KKKKKKK__KKK_KKKK__

(`D`: Du, `K`: Kund:in, `_`: Lücke)

Die Lösung wäre hier: 4.

Für diese Schlange könntest Du, wenn Du jedes Mal an 10 Positionen vorbeilaufen kannst, mit 4 Mal Vordrängeln bis zur vordersten Lücke kommen:

1. Von 0 zu 8
2. Von 8 zu 17
3. Von 17 zu 26
4. Von 26 zu 35

2.4 Aufgabe

Schreibe ein Programm, welches bestimmen kann, wie viele „Vordrängler“ du mindestens brauchst, um zur vordersten Lücke in der Schlange zu kommen. Du kannst dein Programm an der Beispieldatei testen. Dein Programm soll aber auch für beliebige andere Warteschlangen funktionieren. Gib abschließend an, wie viele Vordrängel-Aktionen mindestens für die Datei Schlange.txt notwendig sind.