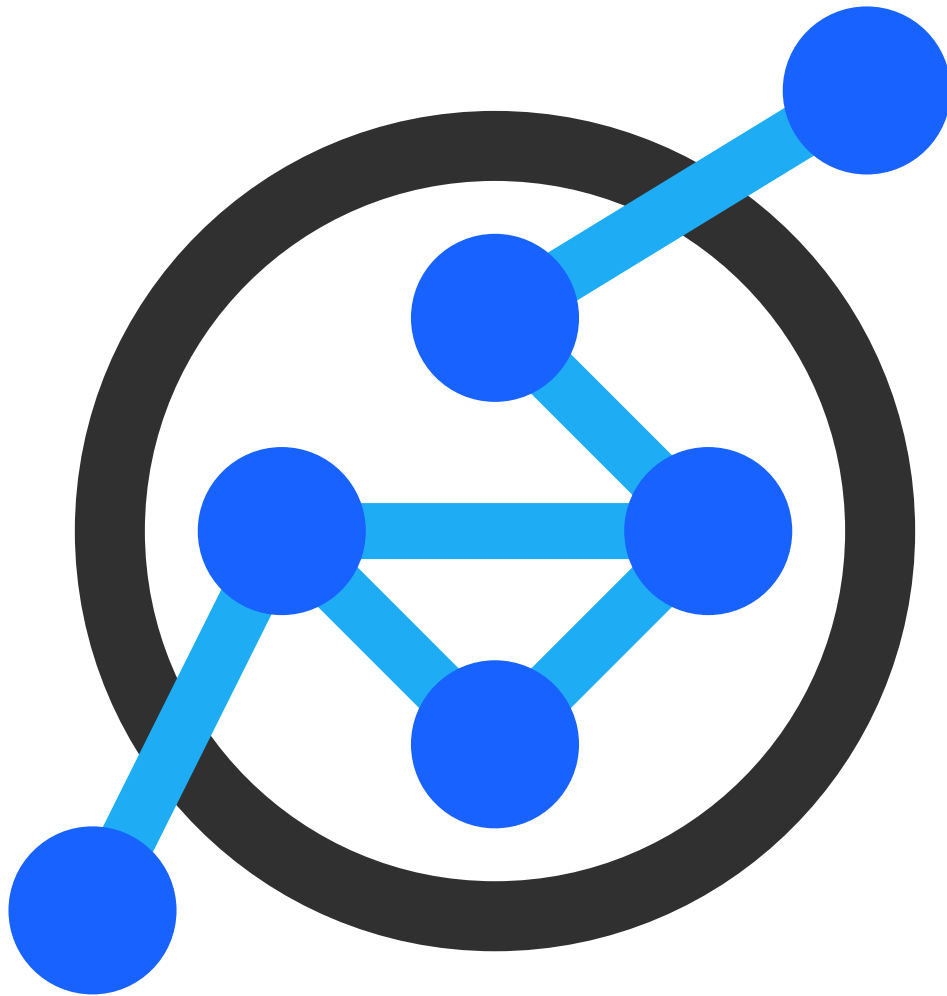


Round 1 SOI 2026/2027

Lösungen



Swiss Olympiad in Informatics

01 September 2026 – 25 September 2026



Bücherstapel

Aufgabenvorbereitung

Übersetzung (de)

Übersetzung (fr)

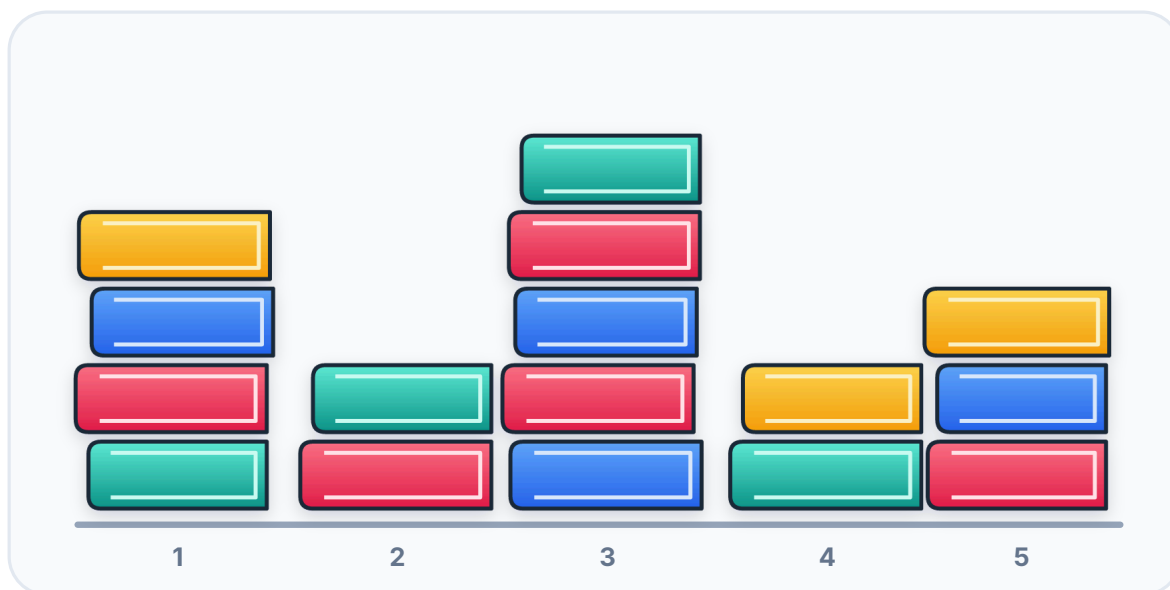
Lösung Autor

Mark Neumann

Benjamin Schmid

Cheng Zhong

Benjamin Schmid



Teil 1: Für seine Maturaprüfungen muss Maus Stofl 5 Bücherstapel lesen, wie im Bild dargestellt. Er stellt fest, dass die fünf Bücherstapel unterschiedlich hoch sind, was ihn stört. Wie viele Bücher müsste Maus Stofl mindestens hinzufügen, um alle Stapel auf die gleiche Höhe zu bringen? Er kann keine Bücher entfernen, da er sie alle lesen muss.

- ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10

Teil 2: Maus Stofl hat erkannt, dass er auch Bücher von einem Stapel zu einem anderen verschieben kann. Wie viele Bücher müssten mindestens verschoben oder hinzugefügt werden, um alle Stapel auf die gleiche Höhe zu bringen?

- ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6



Teilaufgabe 1: Bücher hinzufügen

Wir müssen 9 Bücher hinzufügen.

In diesem Teil können wir nur Bücher hinzufügen. Wir können keine Bücher entfernen oder verschieben. Daher wird ein Stapel nie tiefer, als er anfangs war. Daher ist das Beste, was wir tun können, den höchsten Stapel zu finden und auf allen anderen Stapeln Bücher hinzuzufügen, um die Höhe des höchsten Stapels zu erreichen.

Der höchste Stapel ist Stapel 3 mit 5 Büchern. Stapel 1 fehlt 1 Buch, Stapel 2 fehlen 3 Bücher, Stapel 4 fehlen 3 Bücher und Stapel 5 fehlen 2 Bücher. Insgesamt müssen wir also 9 Bücher hinzufügen.

Teilaufgabe 2: Bücher verschieben

Wir müssen 5 Aktionen ausführen.

Im zweiten Teil können wir auch Bücher verschieben. Wie wir im ersten Teil gesehen haben, wird die Anzahl der Bücher, die wir hinzufügen müssen, durch den höchsten Stapel bestimmt. Daher wollen wir, dass der höchste Stapel so klein wie möglich ist. Wir erreichen dies, indem wir die Bücher gleichmässig auf die Stapel verteilen. Der höchste Stapel nach der Verteilung sagt uns dann, wie hoch jeder Stapel sein wird. Beachte, dass der höchste Stapel höchstens ein Buch höher ist als der niedrigste Stapel, wenn wir die Bücher gleichmässig verteilen.

Insgesamt haben wir 16 Bücher. Mit nur 3 Büchern pro Stapel können wir nur 15 Bücher haben. Daher brauchen wir 4 Bücher pro Stapel (was insgesamt 20 Bücher ermöglichen würde). Um dies zu erreichen, können wir zum Beispiel ein Buch von Stapel 3 zu Stapel 5 verschieben und dann jeweils zwei Bücher auf Stapel 2 und 4 hinzufügen. Dies ergibt insgesamt 5 Aktionen.



Kürzester Weg

Aufgabenvorbereitung

Übersetzung (de)

Übersetzung (fr)

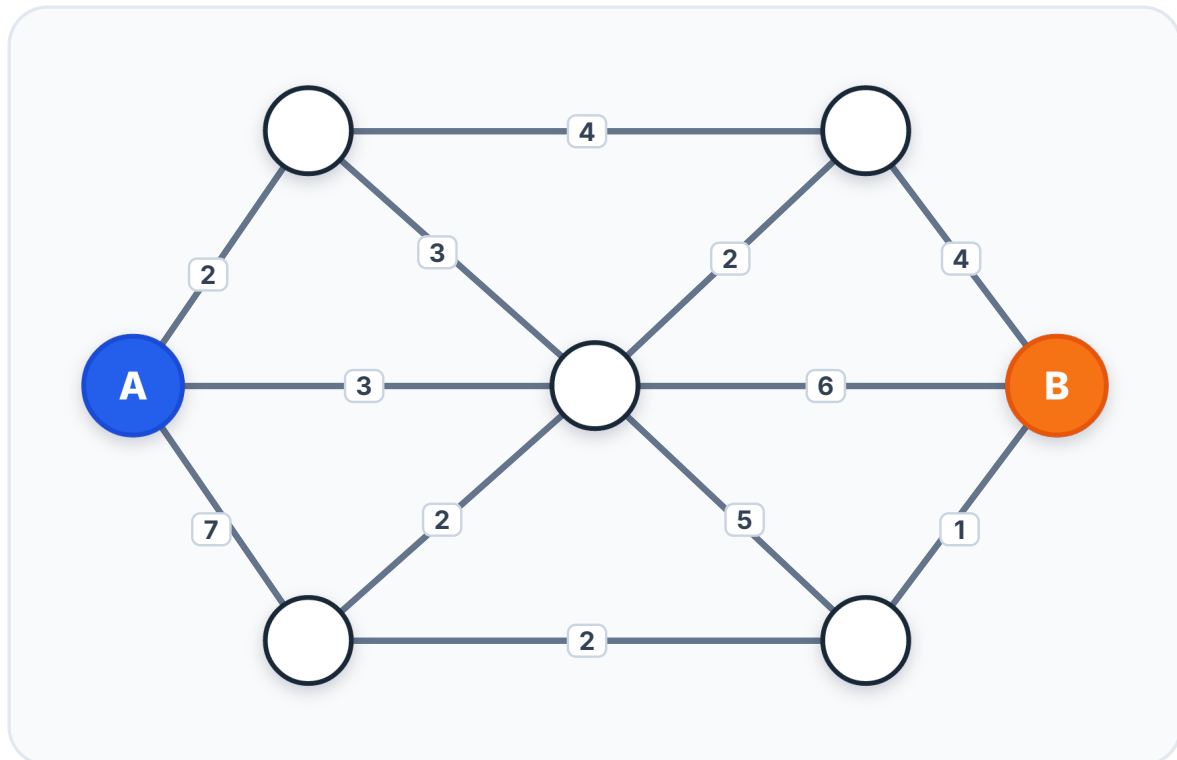
Lösung Autor

Mark Neumann

Benjamin Schmid

Cheng Zhong

Benjamin Schmid



Teil 1: Maus Binna lebt momentan in Stadt A und muss so schnell wie möglich in die Stadt B. Einige der Städte sind durch Strassen verbunden, die jeweils in beide Richtungen befahren werden können. Für jede Strasse wissen wir, wie lange es dauert, die Strasse zu befahren. Im Bild ist eine Karte des Netzes aus Städten und Strassen (mit den entsprechenden Fahrzeiten) dargestellt. Wie lange ist die kürzestmögliche Reisezeit zwischen Stadt A und Stadt B?

- ☐ 7 ☐ 8 ☐ 9 ☐ 10 ☐ 11

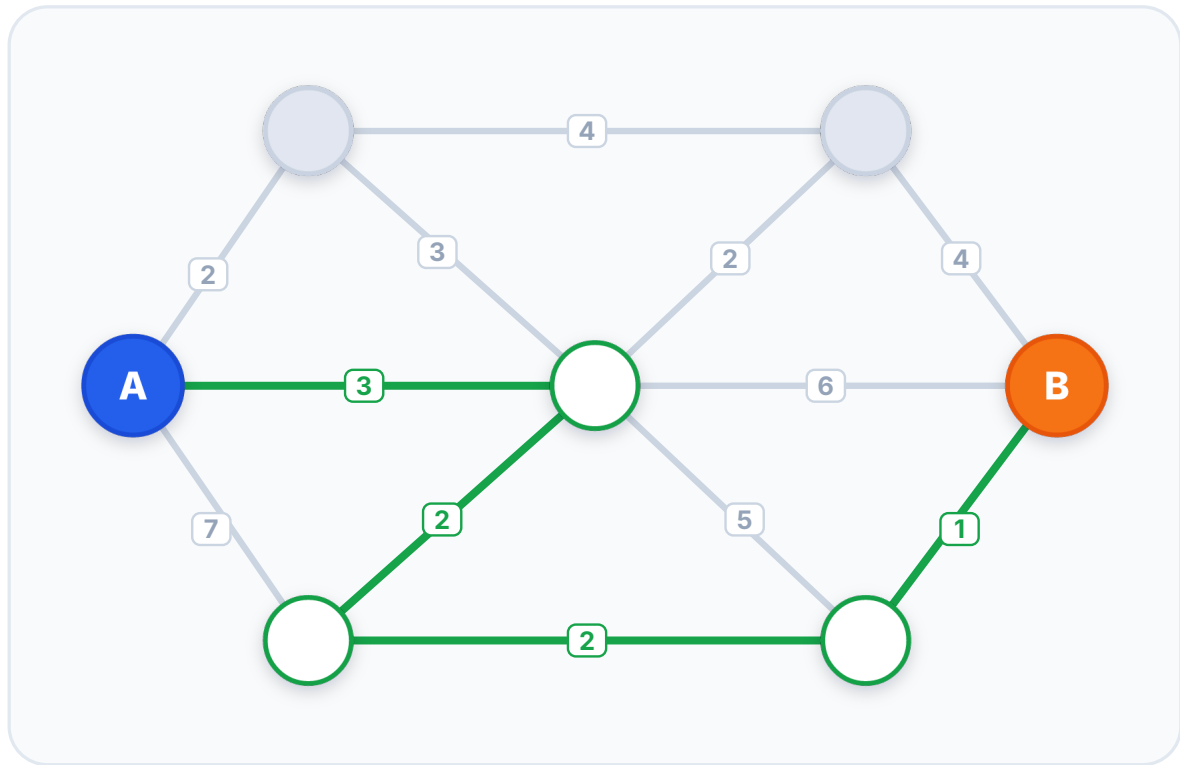
Teil 2: Aufgrund eines Sturms kann es sein, dass maximal eine der Strassen gesperrt wird. Wie viel Zeit muss Maus Binna mindestens einplanen, damit sie unabhängig davon, welche Strasse gesperrt wird, sicher von Stadt A nach Stadt B gelangen kann? Beachte, dass Binna vor Beginn ihrer Reise weiss, welche Strasse gesperrt ist.

- ☐ 8 ☐ 9 ☐ 10 ☐ 11 ☐ 12



Teilaufgabe 1: Direkter Weg

Die kürzest mögliche Reisezeit beträgt 8.

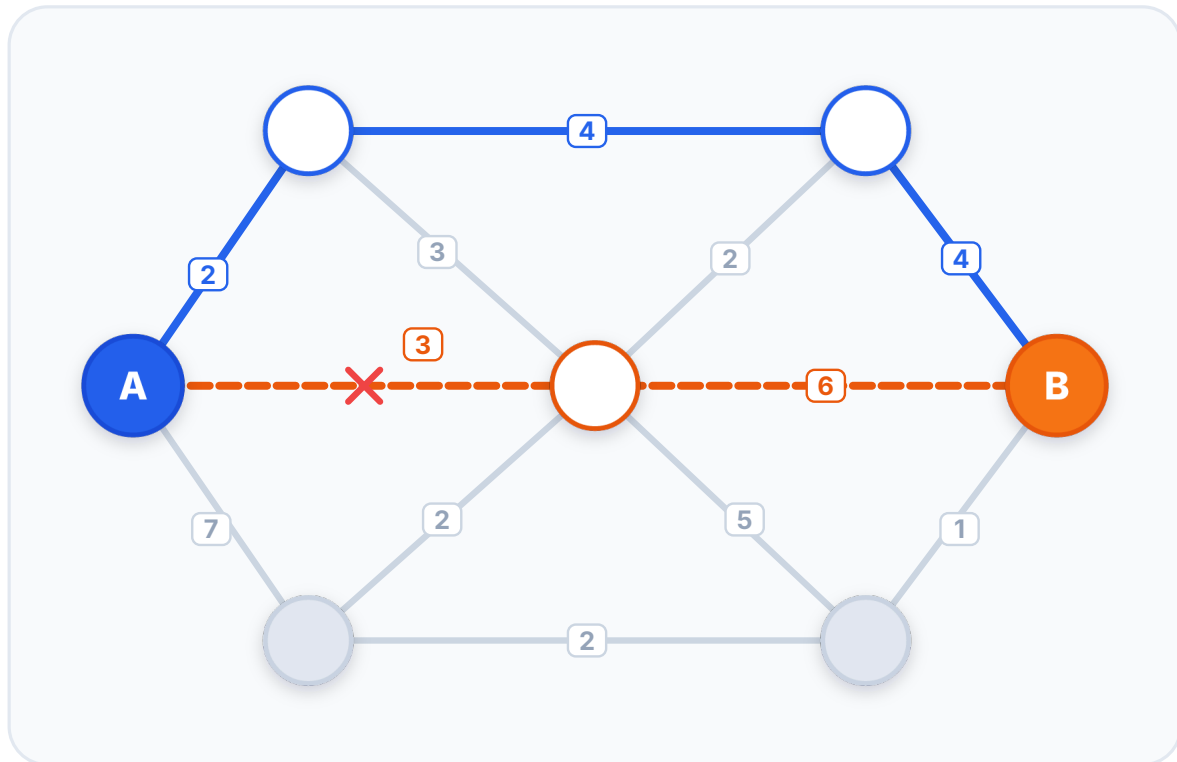


In der Informatik wird diese Art von Netzwerk mit Knoten (= Städte) und Kanten (= Strassen) als Graph bezeichnet. Ein Graph zeigt einige Objekte (Knoten; in unserem Fall Städte) und ihre Verbindungen (Kanten; in unserem Fall Strassen). Eine Kante kann ein Gewicht haben, um eine Art „Entfernung“ anzugeben, die diese Verbindung darstellt.

Um den kürzesten Weg in einem Graphen zu finden, können wir einen von mehreren Algorithmen verwenden, die dieses Problem lösen. Der bekannteste davon ist Dijkstras Algorithmus.

Teilaufgabe 2: Blockierte Strassen

Maus Binna muss mit einer Reisezeit von mindestens 10 planen.



Da Maus Binna die betroffene Strasse vorher kennt, können wir verschiedene Fälle simulieren, welche Strasse betroffen sein könnte. Eine erste Beobachtung ist, dass sich ihre Reisezeit nicht erhöht, wenn die betroffene Strasse nicht auf dem kürzesten Weg aus Teil 1 liegt. Daher müssen wir nur Fälle betrachten, in denen eine der Strassen auf dem kürzesten Weg nicht verfügbar ist. Wir können jede Kante des kürzesten Weges nacheinander entfernen und die kürzeste Route neu berechnen. Von all diesen Routen ist die Lösung das Maximum dessen, was wir gefunden haben.

Wenn wir die Kante der Länge 3 auf der linken Seite entfernen (im Bild oben durchgestrichen gezeigt), hat der kürzeste Weg, den wir finden, die Länge 10, wobei wir immer die oberste Kante nehmen (blauer Weg). Wenn wir eine der anderen Kanten des ursprünglichen kürzesten Weges entfernen, können wir immer den Weg geradeaus mit einer Länge von 9 nehmen (oranger Weg).



Mäuse-Labyrinth

Aufgabenvorbereitung

Übersetzung (de)

Übersetzung (fr)

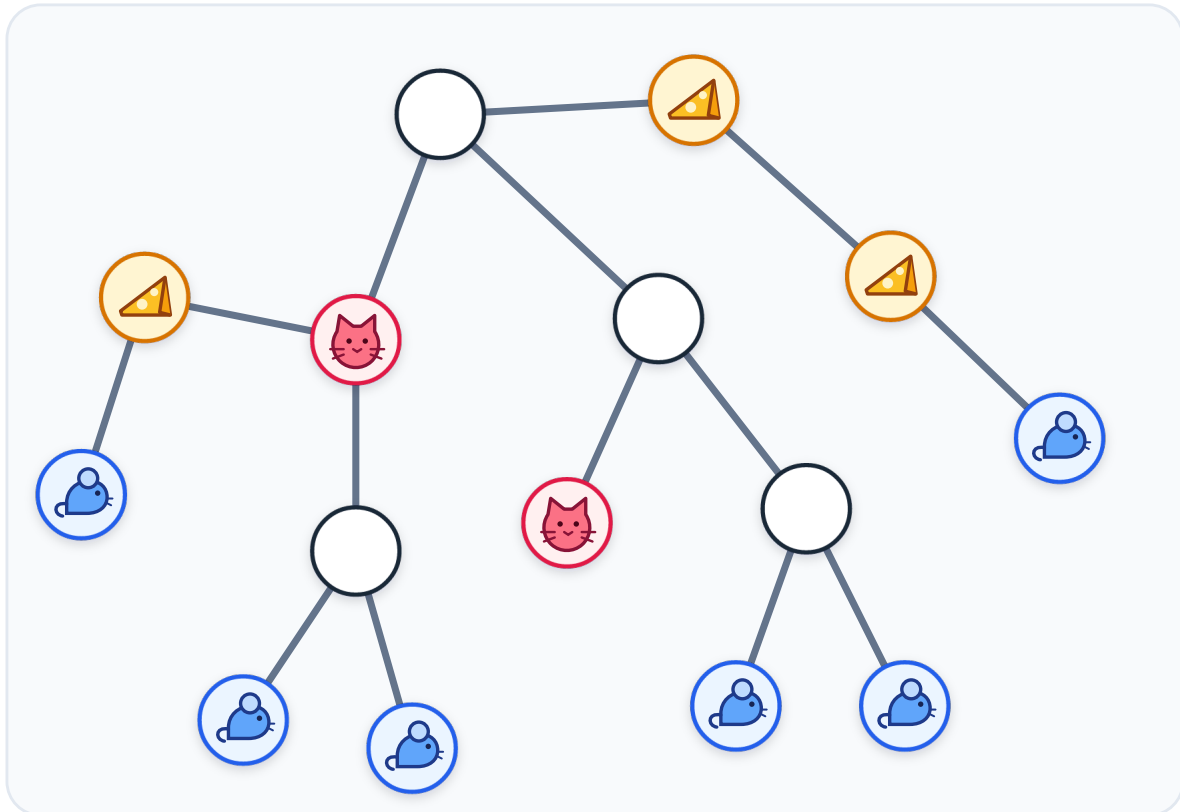
Lösung Autor

Mark Neumann

Benjamin Schmid

Cheng Zhong

Benjamin Schmid



Teil 1: Du hast gerade ein Mäuselabyrinth gebaut. Das Labyrinth besteht aus verschiedenen Räumen, die durch Gänge miteinander verbunden sind, wie im Bild dargestellt.

Einige Räume im Labyrinth enthalten Mäuse, andere enthalten Käsebrocken. Einige Räume werden von einer Katze bewacht, welche die Mäuse nicht passieren können. In der dargestellten Anordnung: Wie viele verschiedene (Maus, Käse)-Paare gibt es, bei denen die Maus erfolgreich den Käse erreichen kann, ohne durch einen von einer Katze bewachten Raum zu gehen? Eine einzelne Maus kann mit mehreren Käsebrocken gepaart werden und umgekehrt.

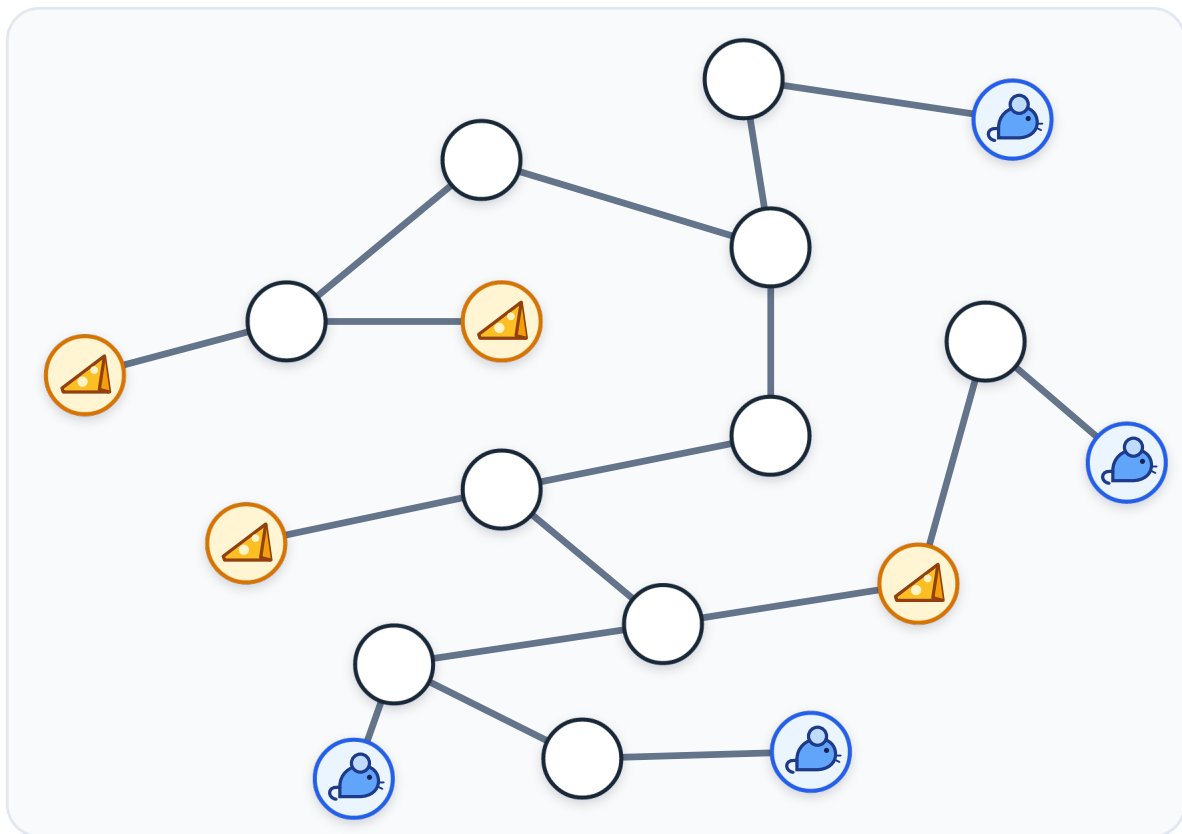
☐ 4

☐ 6

☐ 7

☐ 10

☐ 12



Teil 2: Die Snackzeit ist vorbei, und jetzt möchtest Du verhindern, dass die Mäuse an den Käse gelangen. Das Labyrinth sieht nun wie im Bild aus.

Wie viele Räume müssen mindestens mit einer Katze besetzt werden, damit keine Maus ein Stück Käse erreichen kann? Du kannst keine Katze in einem Raum mit einer Maus oder mit Käse platzieren.

- ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6



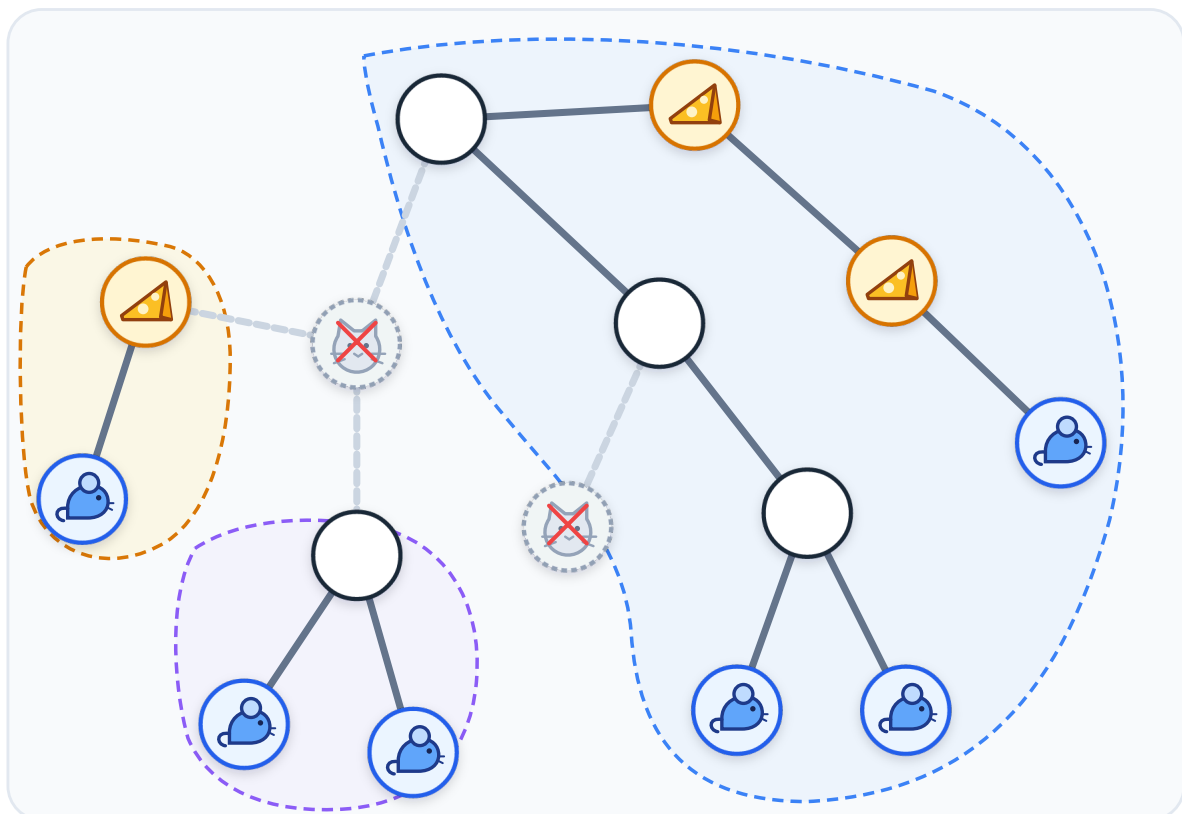
Teilaufgabe 1: Zähle die Paare

Es gibt 7 (Maus, Käse)-Paare, bei denen die Maus den Käse erreichen kann.

Diese Darstellung des Labyrinths mit Knoten (= Räume) und Kanten (= Gänge) wird als Graph bezeichnet. Im ersten Teil interessieren wir uns für Komponenten im Graphen. Eine Komponente ist eine Menge von Knoten des Graphen, die durch Kanten verbunden sind, entweder direkt oder indirekt.

Anfangs gibt es nur eine Komponente: Wenn wir die Katzen ignorieren, können wir von jedem Knoten jeden anderen Knoten erreichen. Jedoch wissen wir, dass ein Knoten, auf dem eine Katze sitzt, nicht passiert werden kann. Daher entfernen wir ihn aus dem Graphen. Dies führt dazu, dass der Graph in mehrere Komponenten zerfällt. Eine Maus in einer Komponente kann jeden Käse in derselben Komponente erreichen, aber sie kann keinen Käse in einer anderen Komponente erreichen. Daher können wir jede Komponente einzeln betrachten und die Anzahl Mäuse mit der Anzahl Käse in dieser Komponente multiplizieren, um die Anzahl Paare in dieser Komponente zu erhalten. Dann addieren wir alle Komponenten, um die Gesamtzahl zu erhalten.

Das Ergebnis ist also $1 \cdot 1 + 2 \cdot 0 + 3 \cdot 2 = 1 + 0 + 6 = 7$.

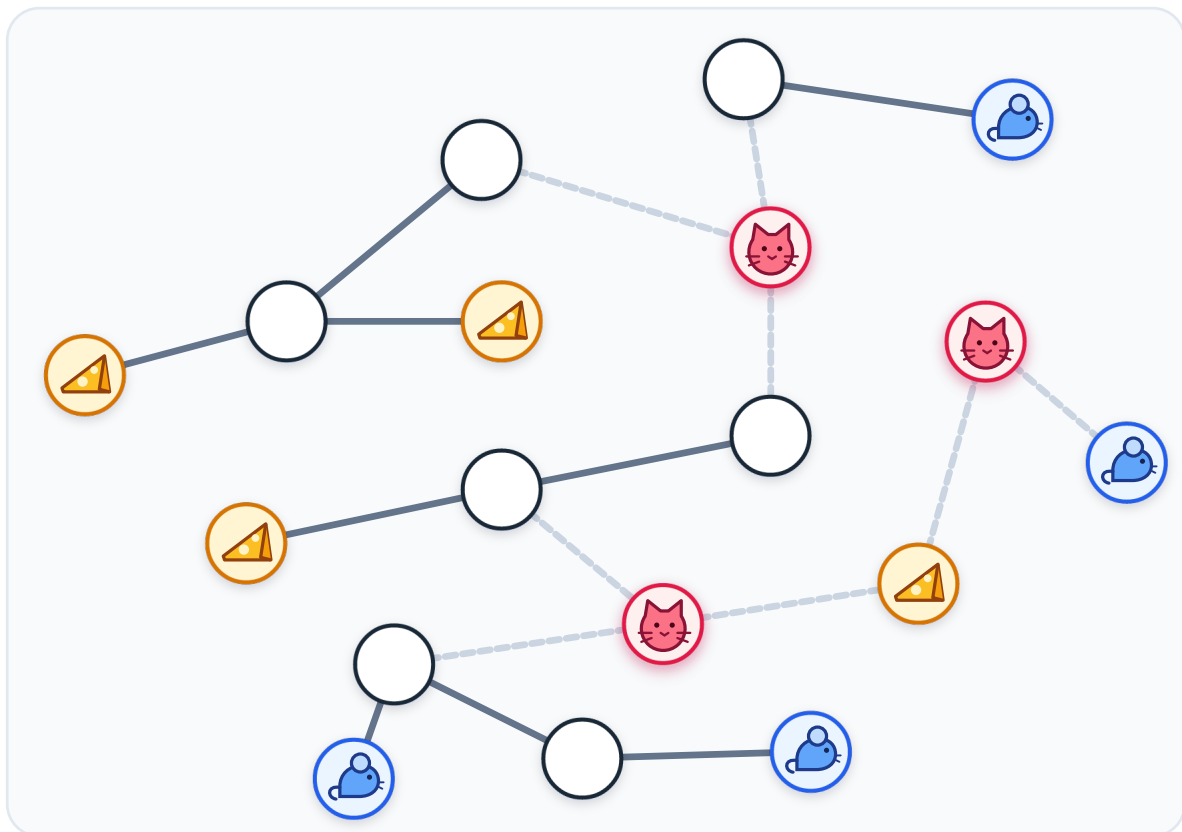


Die drei Komponenten sind mit verschiedenen Farben hervorgehoben: die rechte Komponente (blau) enthält drei Mäuse und zwei Käse ($3 \cdot 2 = 6$ Paare), die linke Komponente (gelb) enthält eine Maus und einen Käse ($1 \cdot 1 = 1$ Paar), und die untere Komponente (lila) enthält zwei Mäuse, aber keinen Käse ($2 \cdot 0 = 0$ Paare).

Teilaufgabe 2: Die Snackzeit ist vorbei

Wir brauchen mindestens 3 Katzen.

Im zweiten Teil müssen wir Knoten auswählen, auf die wir Katzen setzen, sodass die resultierenden Komponenten entweder nur Mäuse oder nur Käse enthalten, aber nicht beides. Es gibt mehrere Möglichkeiten, eine davon ist unten dargestellt.



Die drei Katzen werden auf die im Diagramm rot markierten Räume gesetzt. Dies teilt das Labyrinth in sechs Komponenten auf, die jeweils entweder nur Mäuse oder nur Käse enthalten.

Bei einer so kleinen Problemgrösse kann dies von Hand durchgeführt werden, indem man ein paar verschiedene Knoten ausprobiert. Eine allgemeine Lösung könnte den Max-Flow-Algorithmus verwenden, um den Min-Cut in einem speziell konstruierten Graphen zu finden. Die Erklärung davon ist allerdings ausserhalb des Rahmens dieses Lösungshefts.

Wandern

Aufgabenvorbereitung

Mark Neumann, Benjamin Schmid

Übersetzung (de)

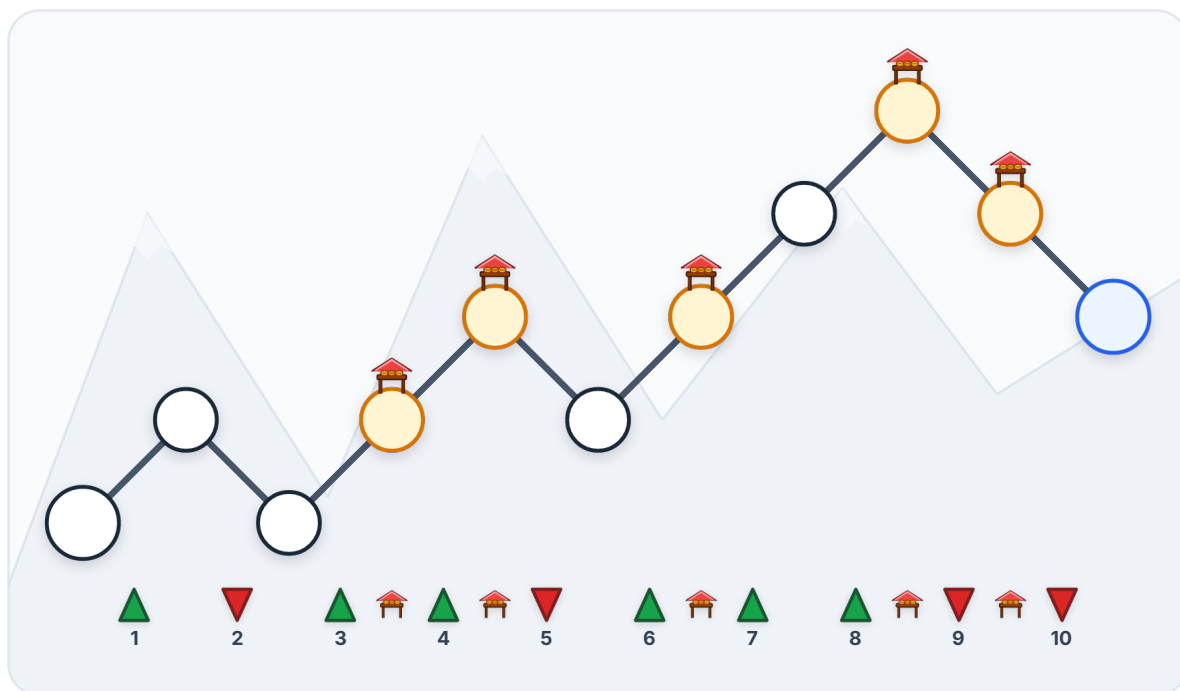
Benjamin Schmid

Übersetzung (fr)

Cheng Zhong

Lösung Autor

Benjamin Schmid



Teil 1: Du möchtest eine Bergwanderung machen und nimmst 3 Sandwiches mit.

Die Wanderung besteht aus 10 Abschnitten und führt an einigen Sandwich-Ständen vorbei, wie auf dem Bild gezeigt. Bevor du einen Abschnitt beschreitest, durchläufst du jeweils diese zwei Schritte:

Schau dir deinen aktuellen Startpunkt für den Abschnitt an.

- Wenn sich dort ein Stand befindet: Du kaufst 2 Sandwiches.
- Wenn sich dort kein Stand befindet: Du kaufst 0 Sandwiches.

Schau dir den Abschnitt an, den du gleich wandern wirst.

- Wenn der Weg bergauf führt und du dich an einem Stand befindest: Du isst 2 Sandwiches.
- Wenn der Weg bergauf führt und du dich nicht an einem Stand befindest: Du isst 1 Sandwich.
- Wenn der Weg bergab führt: Du isst 0 Sandwiches.

Wie viele Sandwiches hast du am Ende der Wanderung noch übrig?

Hinweis: Wenn du nicht genügend Sandwiches hast, musst du deine Wanderung abbrechen.

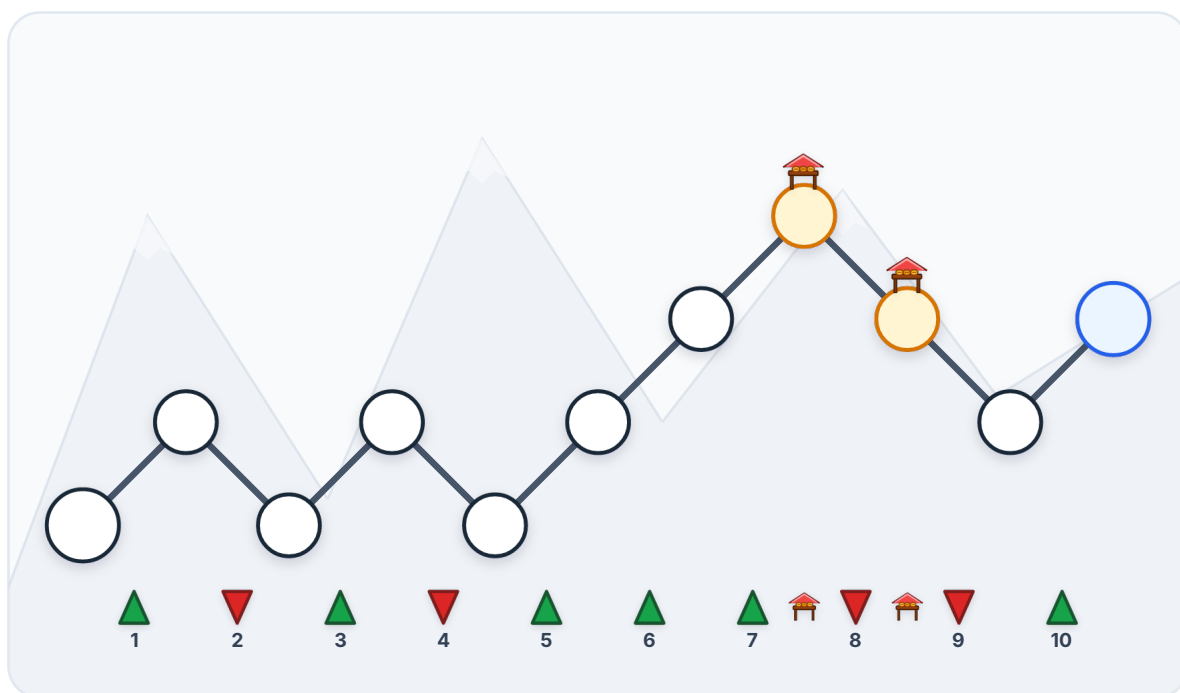
☐ 3

☐ 4

☐ 5

☐ 6

☐ 8



Teil 2: Als Du am Startpunkt ankommst, stellst Du fest, dass Du Deine Sandwiches vergessen hast! Es gibt nur eine Lösung: Hoffen, dass unterwegs neue Stände eröffnen. Neue Stände können am Anfang eines jeden Abschnitts (einschliesslich des ersten) hinzugefügt werden, wo es noch keinen Stand gibt.

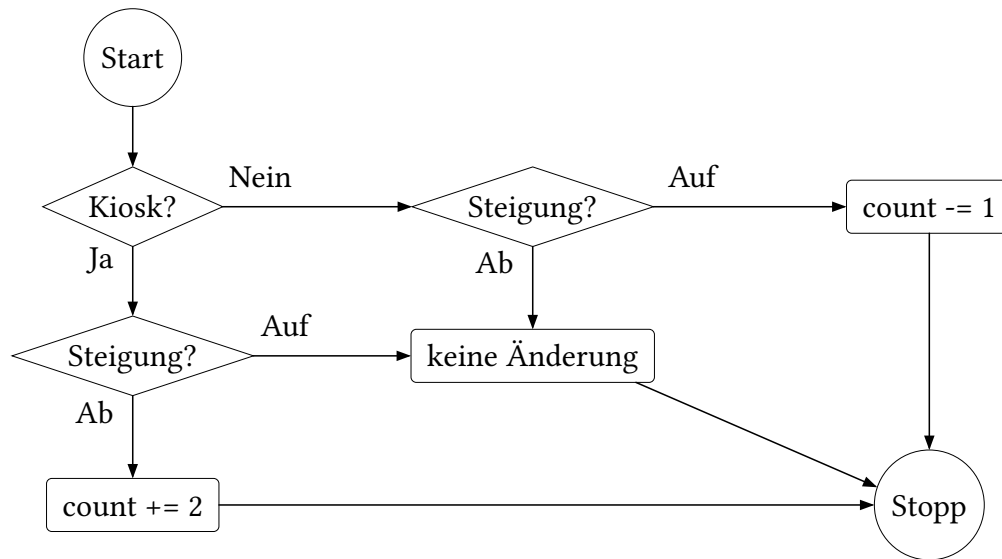
Wie viele neue Stände müssen mindestens hinzugefügt werden, damit Du die gesamte Wanderung gemäss den Regeln erfolgreich absolvieren kannst, obwohl Du mit null Sandwiches gestartet bist?

- ☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4

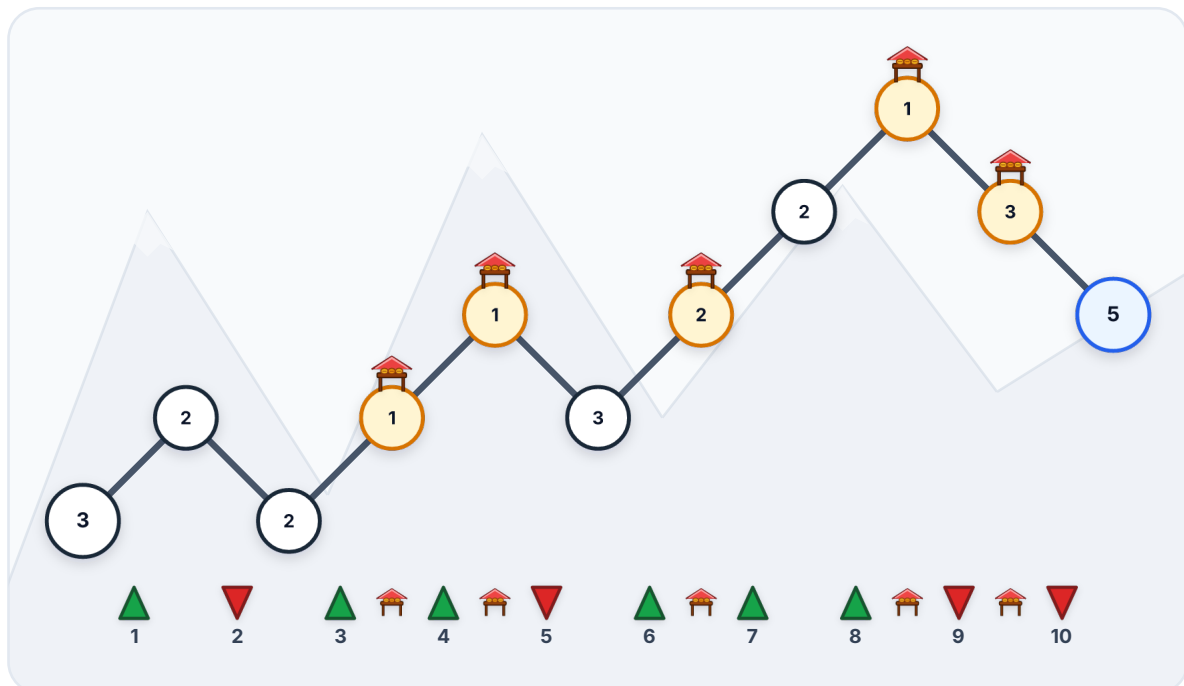
Teilaufgabe 1: Drei Sandwiches

Am Ende der Wanderung haben wir 5 Sandwiches.

Diese Aufgabe besteht darin, einen einfachen Algorithmus auszuführen und das Ergebnis zu ermitteln. Wir können den Algorithmus etwas vereinfachen, indem wir einfach notieren, was mit der Anzahl Sandwiches passiert, abhängig davon, ob es einen Kiosk gibt oder nicht, und abhängig von der Steigung. Wir initialisieren die Anzahl Sandwiches (im Folgenden „count“) mit 3 und führen dann für jeden Abschnitt Folgendes aus:



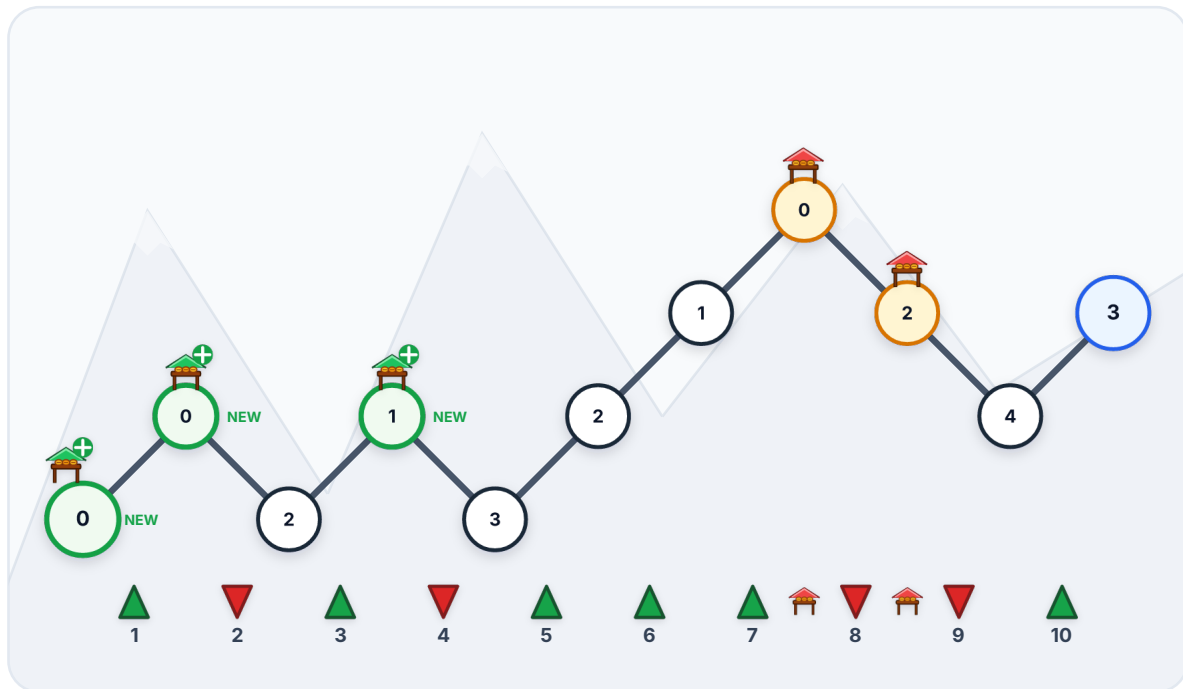
Wir erhalten die folgenden Anzahlen von Sandwiches an den verschiedenen Knoten:





Teilaufgabe 2: Die vergessenen Sandwiches

Wir müssen mindestens 3 Kioske hinzufügen.



In diesem Teil müssen wir herausfinden, wie viele Kioske wir wo platzieren müssen. Wir können einige Beobachtungen machen:

- Wenn wir einen Kiosk vor einem Aufstieg platzieren, können wir den nächsten Abschnitt passieren, aber wir haben danach gleich viele Sandwiches wie zuvor. Dieser Kiosk erlaubt uns also, einen zusätzlichen Aufstieg (den direkt nach dem Kiosk) zu bewältigen.
- Wenn wir einen Kiosk vor einem Abstieg platzieren, haben wir nach dem nächsten Abschnitt zwei Sandwiches mehr, die wir nutzen können, um zwei zusätzliche Aufstiege ohne vorgelagerten Kiosk zu bewältigen.
- Einen Kiosk vor einem Abstieg zu platzieren, ist besser als vor einem Aufstieg, da ersterer uns erlaubt, zwei Aufstiege zu bewältigen, während letzterer uns nur einen ermöglicht.
- Daher sollten wir einen Kiosk nur dann vor einem Aufstieg platzieren, wenn es absolut nötig ist.

Mit diesen Beobachtungen können wir die optimalen Abschnitte für zusätzliche Kioske bestimmen. Wir brauchen einen Kiosk vor dem ersten Abschnitt, da wir keine Sandwiches haben und mit einem Aufstieg beginnen. Vor dem zweiten Abschnitt prüfen wir, ob wir das Ende bereits ohne weitere Kioske erreichen können, was aber nicht der Fall ist. Daher fügen wir auch hier einen Kiosk hinzu, was nach dem zweiten Abschnitt zu zwei Sandwiches führt. Der dritte Abschnitt ist wieder ein Aufstieg, aber wir haben Sandwiches, also fügen wir keinen Kiosk hinzu. Vor dem vierten Abschnitt prüfen wir erneut, ob wir das Ende ohne zusätzliche Kioske erreichen können, was aber immer noch nicht der Fall ist. Daher fügen wir einen Kiosk hinzu und haben nach dem vierten Abschnitt drei Sandwiches. Die nächsten paar Abschnitte sind Aufstiege, aber wir haben genau genug Sandwiches für alle, sodass wir keine weiteren



Kioske hinzufügen müssen. Sobald wir den Gipfel erreicht haben, gibt es zwei Kioske am Abstieg, was ausreicht, um das Ende ohne weitere Kioske zu erreichen. Daher brauchen wir nur drei zusätzliche Kioske.



Solaria

Aufgabenvorbereitung

Übersetzung (de)

Übersetzung (fr)

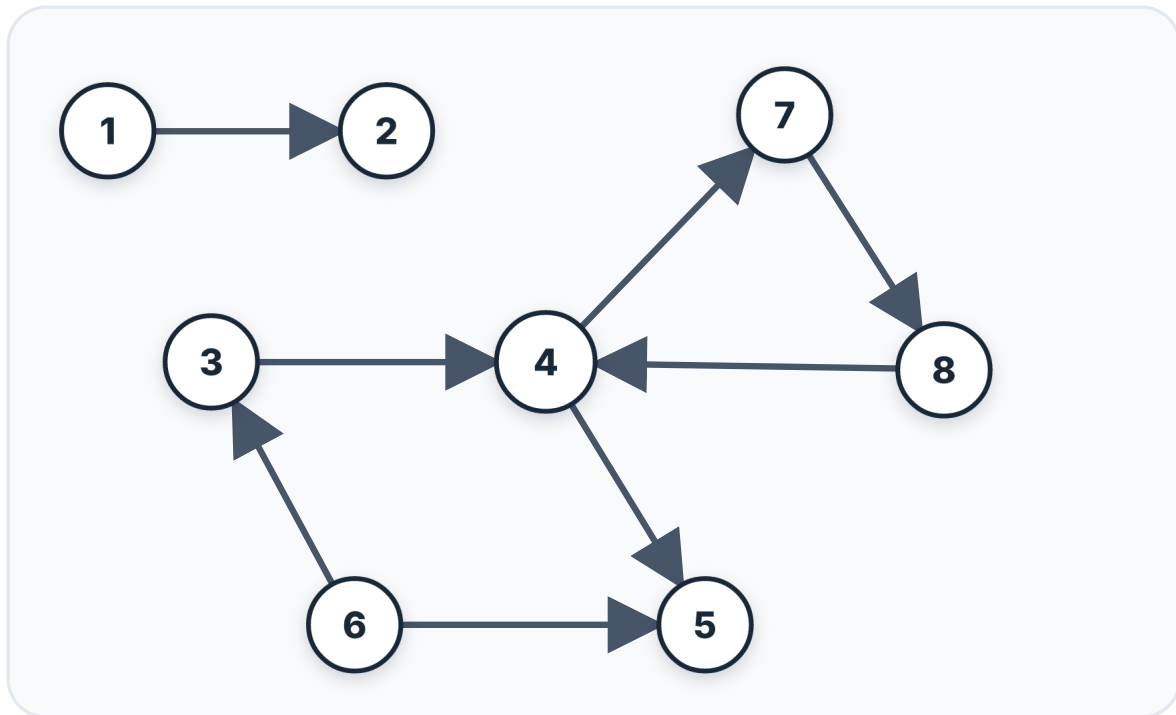
Lösung Autor

Mark Neumann

Benjamin Schmid

Cheng Zhong

Benjamin Schmid

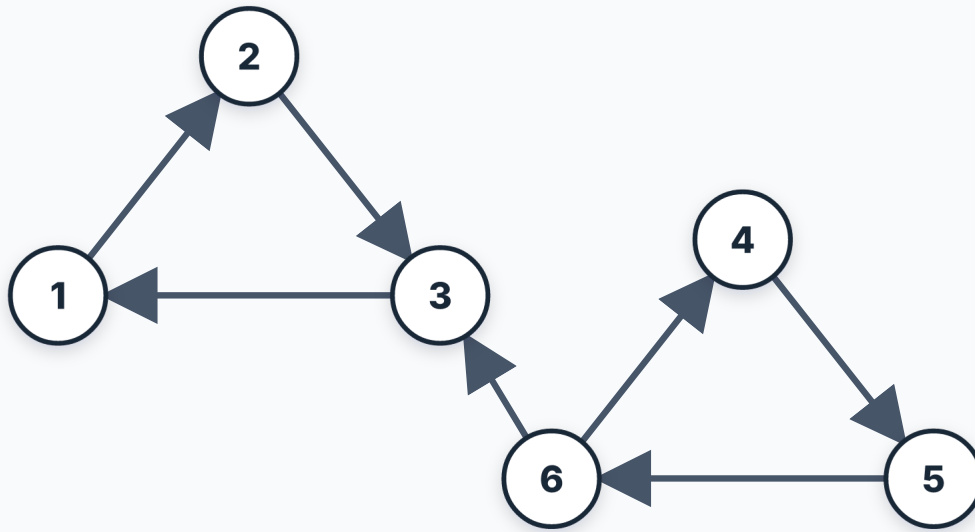


Teil 1: Die Bergstadt Solaria ist auf ein Netz von schmalen Einbahnbrücken angewiesen, um die Bezirke miteinander zu verbinden. Aber die Stadtplaner haben einen schrecklichen Fehler gemacht: Weil die Brücken nur in eine Richtung befahren werden können, werden Autofahrer eingesperrt! Es gibt Teile der Stadt, in die man zwar gelangt, aus denen es aber keinen Weg zurück nach Hause gibt. Der Bürgermeister möchte das Netz sofort reparieren, aber das Stadtbudget kann sich nur wenige neue Brücken leisten.

Die Karte des Brückennetzes im Bild zeigt an, in welche Richtung jede Brücke befahren werden darf. Wie viele neue Einbahnbrücken muss die Stadt mindestens bauen, damit ein Autofahrer in jedem Bezirk starten und erfolgreich zu jedem anderen Bezirk reisen kann?

Hinweis: Es ist erlaubt, dass zwei Brücken zwischen denselben zwei Bezirken in entgegengesetzten Richtungen verlaufen.

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

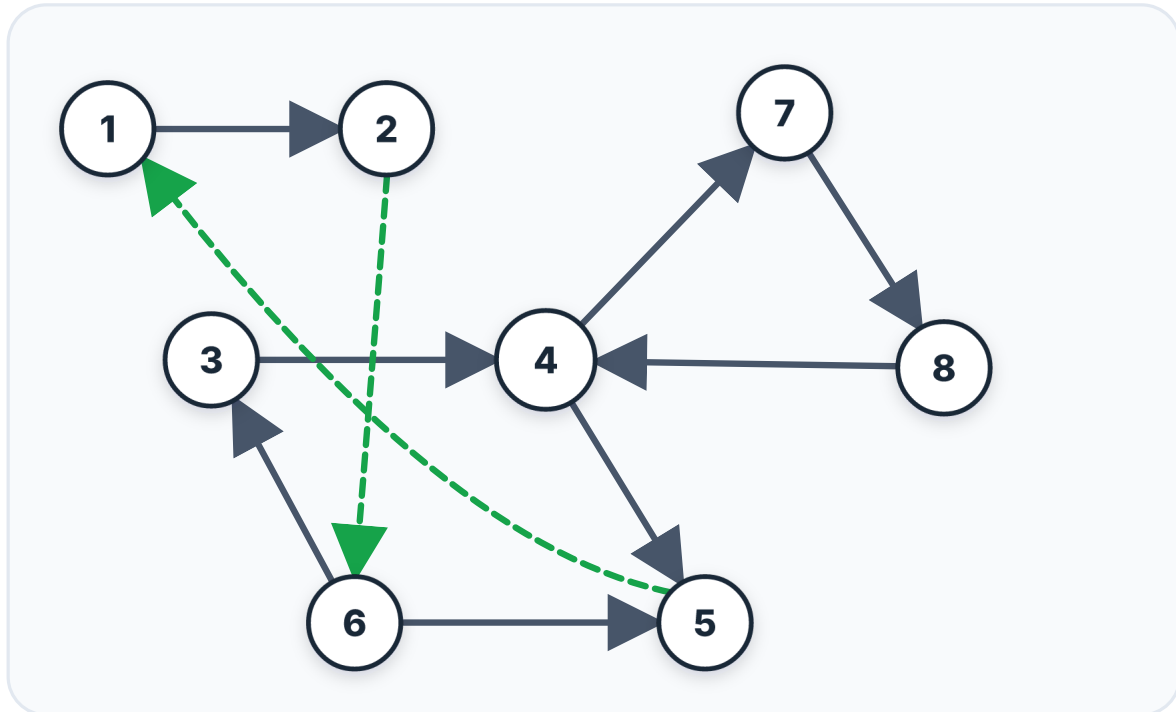


Teil 2: Betrachte die neue Karte. Es gibt mehrere Möglichkeiten, genau eine neue Einbahnbrücke hinzuzufügen, sodass ein Autofahrer in jedem Bezirk starten und erfolgreich zu jedem anderen Bezirk reisen kann. Auf wie viele Arten kann dies gemacht werden?

- ☐ 3 ☐ 5 ☐ 6 ☐ 8 ☐ 9

Teilaufgabe 1: Neue Brücken

Wir können dies erreichen, indem wir nur zwei Kanten hinzufügen: $5 \rightarrow 1$ und $2 \rightarrow 6$.



Die Darstellung des Brückennetzes mit Knoten (= Bezirke) und Kanten (= Brücken) wird als Graph bezeichnet. In diesem Fall können die Kanten nur in eine Richtung benutzt werden, daher nennen wir dies einen gerichteten Graphen, welcher gerichtete Kanten hat.

Für diese Aufgabe ist das Konzept einer „stark zusammenhängenden Komponente“, kurz SCC (strongly connected component), hilfreich. Eine SCC ist eine Menge von Knoten, sodass wir von jedem Knoten in der Menge jeden anderen Knoten in der Menge erreichen können. Anfangs gibt es eine SCC mit den Knoten 4, 7 und 8. Das Ziel der Aufgabe ist es, eine grosse SCC mit allen Knoten zu schaffen, während wir möglichst wenige neue Kanten hinzufügen.

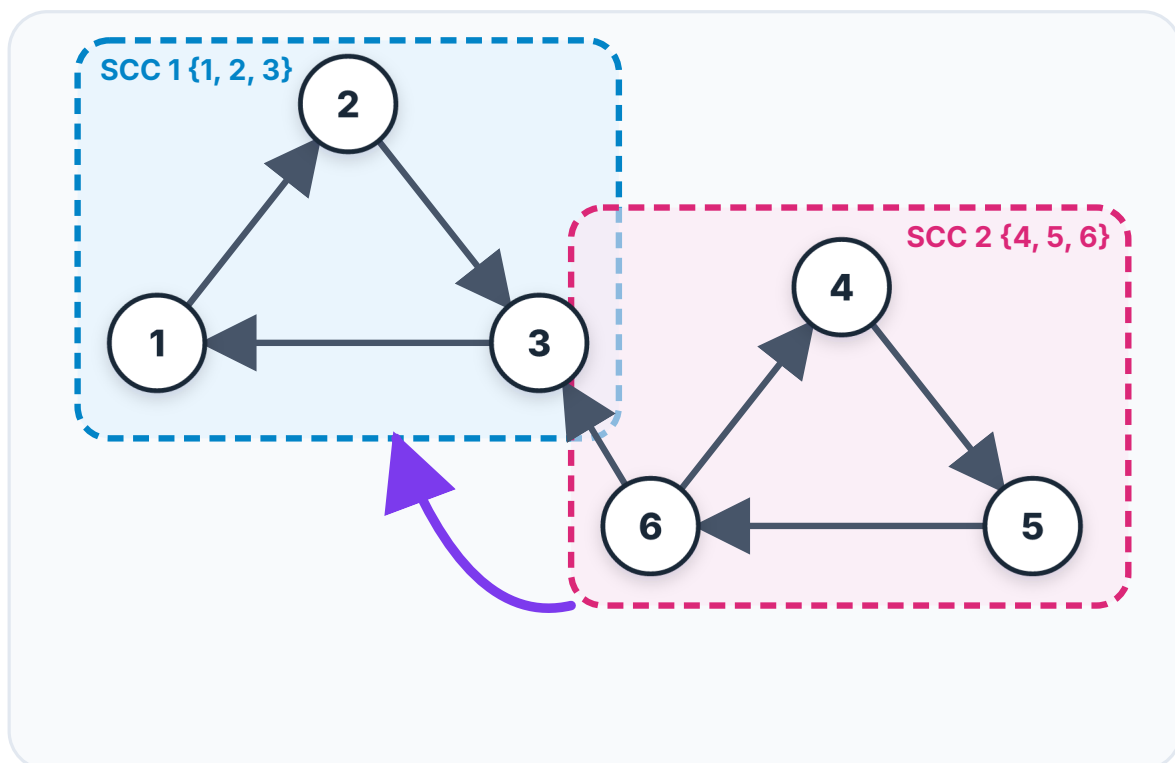
Wir müssen eine ausgehende Kante zu jedem Knoten hinzufügen, der noch keine ausgehende Kante hat, und wir müssen eine eingehende Kante zu jedem Knoten hinzufügen, der noch keine eingehende Kante hat. Die Knoten ohne ausgehende Kante sind anfangs die Knoten 2 und 5. Die Knoten ohne eingehende Kante sind anfangs die Knoten 1 und 6. Wenn wir sie so paaren, dass die kleine „Insel“ aus 1 und 2 mit dem Rest verbunden ist, können wir eine grosse SCC bilden.

Beachte, dass dies nicht allgemein gilt. Es gibt Beispiele, bei denen alle Knoten eingehende und ausgehende Kanten haben, der Graph aber dennoch keine grosse SCC ist. Der Graph in Teil 2 ist ein solches Gegenbeispiel.

Teilaufgabe 2: Eine neue Brücke

Es gibt 9 Möglichkeiten, wie wir eine Kante hinzufügen können, um die Bedingung zu erfüllen.

Wie im ersten Teil wollen wir eine grosse SCC erschaffen. Derzeit hat der Graph zwei kleine SCCs: eine mit den Knoten 1, 2 und 3, und eine mit den Knoten 4, 5 und 6. Da wir jeden anderen Knoten innerhalb einer SCC erreichen können, können wir eine SCC fast als einen Knoten betrachten und alle Knoten in der SCC durch einen einzigen Knoten ersetzen. Der Graph würde dann sehr einfach werden, mit nur zwei Knoten, einen für die erste SCC und einen für die zweite SCC, mit einer Kante von der zweiten SCC zur ersten SCC.



Von einem Knoten in der zweiten SCC aus können wir bereits jeden anderen Knoten im Graphen erreichen. Es ist jedoch nicht möglich, von einem Knoten in der ersten SCC zu einem Knoten in der zweiten SCC zu gelangen. Dies ist auch in unserem vereinfachten Graphen mit nur zwei Knoten deutlich sichtbar.

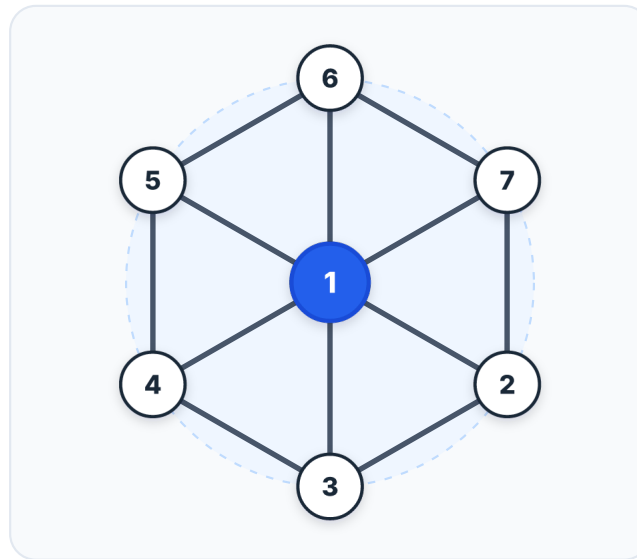
Daher müssen wir eine Kante von der ersten SCC zur zweiten SCC hinzufügen. Da wir jeden anderen Knoten innerhalb einer SCC erreichen können, spielt es keine Rolle, welchen Knoten wir innerhalb einer SCC wählen. Daher können wir eine Kante von jedem Knoten in der ersten SCC zu jedem Knoten in der zweiten SCC hinzufügen. Es gibt drei Knoten zur Auswahl in der ersten SCC und drei Knoten in der zweiten SCC. Somit haben wir $3 \cdot 3 = 9$ mögliche Kanten, die wir hinzufügen können, um den Graphen zu einer grossen SCC zu vereinen.



Atlantis

Aufgabenvorbereitung
Übersetzung (de)
Übersetzung (fr)
Lösung Autor

Mark Neumann
Benjamin Schmid
Cheng Zhong
Benjamin Schmid



Teil 1: Das mythische Land Atlantis hat ein eigenartiges Eisenbahnnetz, das seine n Städte verbindet. Es besteht aus einem zentralen Halt, der mit jeder anderen Stadt durch eine direkte Bahnlinie verbunden ist. Die restlichen $n - 1$ Städte bilden einen Ring von Bahnlinien um die zentrale Stadt. Ein Beispiel für $n = 7$ ist dargestellt. Formal gibt es eine direkte Bahnlinie zwischen Stadt 1 und jeder anderen Stadt. Es gibt auch eine direkte Bahnlinie von Stadt x nach $x + 1$ für alle $2 \leq x \leq n - 1$ und eine direkte Bahnlinie von Stadt 2 nach n . Jede Bahnlinie kann in beide Richtungen befahren werden.

Maus Stofl möchte Atlantis erkunden, indem er an einer Station startet und entlang des Schienennetzes reist. Um die Sache interessant zu halten, möchte er jedoch jede Bahnlinie höchstens einmal benutzen. Wie viele Bahnlinien kann Maus Stofl für $n = 7$ während seiner Tour durch Atlantis in einer zusammenhängenden Reihenfolge höchstens befahren, ohne eine Bahnlinie zweimal zu nutzen?

Ein Beispiel einer gültigen Tour, die Stofl nehmen kann, ist $1 \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow 5$. Dieser Weg besteht aus 4 Bahnlinien.

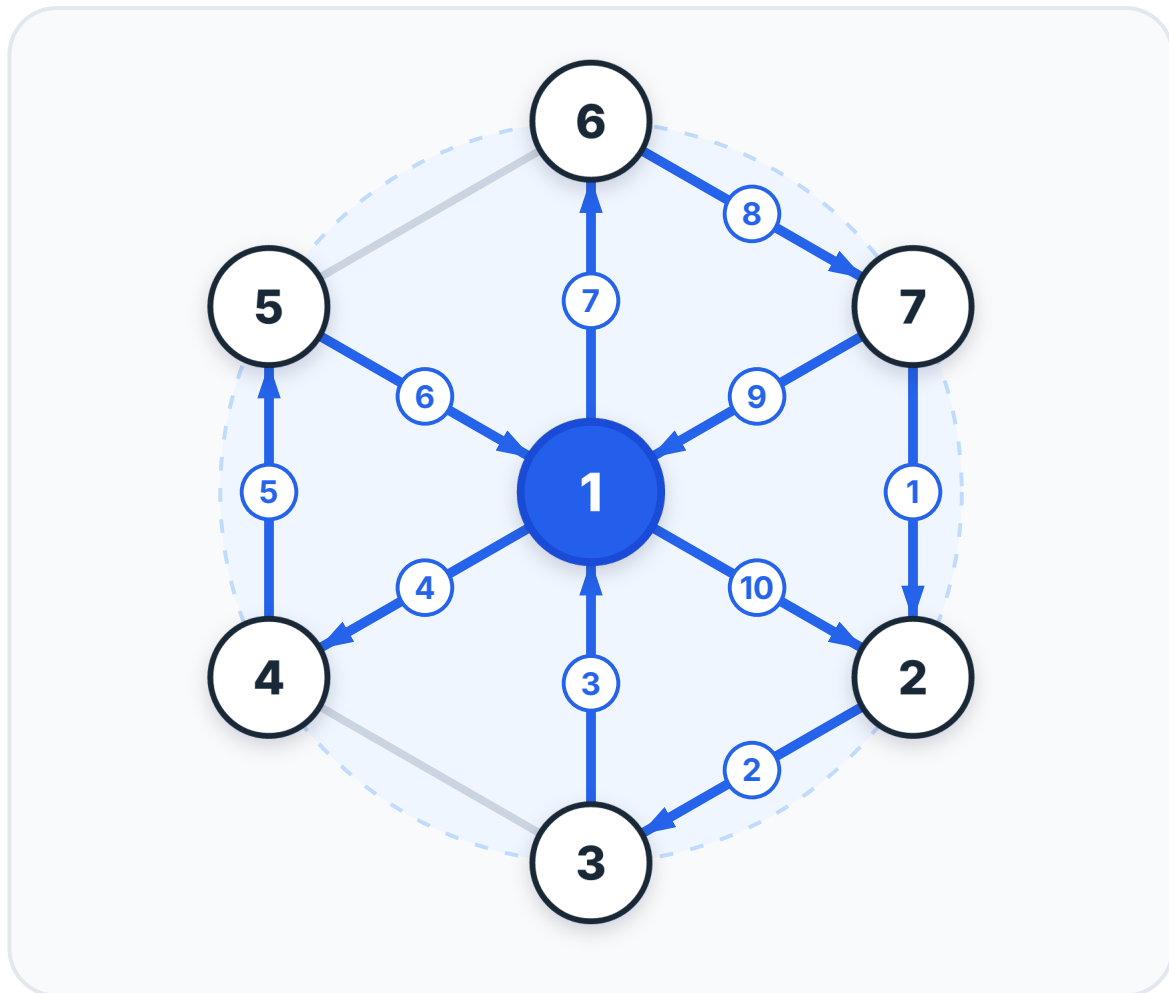
☐ 7 ☐ 8 ☐ 9 ☐ 10 ☐ 11

Teil 2: Nehmen wir nun an, Atlantis sei viel grösser und $n = 2027$. Wie lang ist die längste Tour, die Stofl jetzt nehmen kann?

.....

Teilaufgabe 1: Kleines Land

Die längste Tour, die wir finden können, hat die Länge 10. Es gibt mehrere solcher Touren, ein Beispiel ist $7 \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow 4 \rightarrow 5 \rightarrow 1 \rightarrow 6 \rightarrow 7 \rightarrow 1 \rightarrow 2$.



Die Darstellung des Schienennetzes mit Knoten (= Städte) und Kanten (= Schienenlinien) wird als Graph bezeichnet.

Es gibt verschiedene Möglichkeiten, wie wir uns in einem Graphen „bewegen“ können. Die einfachste nennt sich Weg (engl. walk), bei dem wir einfach von einem Knoten zu einem seiner Nachbarn gehen, die durch eine Kante verbunden sind. Wir können jeden Knoten und jede Kante mehrmals verwenden. Wenn wir den Weg einschränken, sodass wir einen Knoten höchstens einmal besuchen, nennen wir das einen Pfad. In unserem Fall sind die Knoten nicht eingeschränkt, aber wir können jede Kante höchstens einmal verwenden. Dies nennt man einen Kantenzug (engl. trail).

Die Aufgabe ist es, den längsten möglichen Kantenzug in einem Graphen zu finden. Im Allgemeinen gibt es keinen effizienten Algorithmus, um den längsten Kantenzug zu finden, aber wir können die spezielle Form des Graphen nutzen, um eine Lösung zu finden. Da der



Graph klein ist, können wir einfach verschiedene Kantenzüge oder Muster ausprobieren, wie wir den Kantenzug bilden.

Teilaufgabe 2: Grosses Land

Die längste Tour hat die Länge 3040.

Da der Graph zu gross ist, um eine solche Tour von Hand zu finden, müssen wir ein Muster erkennen. Betrachten wir die Lösung aus Teil 1, sehen wir, dass die Tour meist von der Mitte zu einem äusseren Knoten, zu einem Nachbarn und dann zurück zur Mitte führt. Es gibt eine kleine Änderung des Musters am Anfang und Ende der Tour. Ohne den Spezialfall am Anfang und Ende wäre unsere Tour einfach die Dreiecke, die durch das Muster gebildet werden.

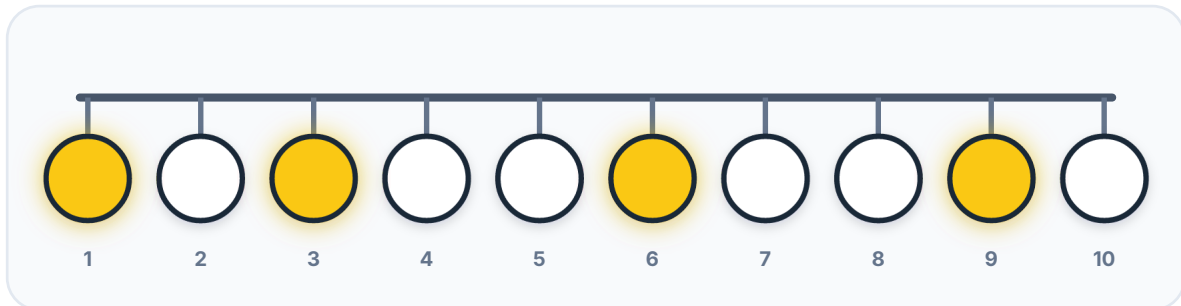
Wie viele solcher Dreiecke passen in den Graphen, ohne dass sich Kanten überschneiden? Jedes Dreieck benötigt eine äussere Kante, und wir können jede zweite äussere Kante für ein Dreieck verwenden. Ein Dreieck fügt 3 zu unserer Tourlänge hinzu. Mit n Knoten haben wir $n - 1$ äussere Kanten. Daher ergibt das normale Muster eine Tour der Länge $\frac{n-1}{2} \cdot 3$.

Durch die spezielle Behandlung von Anfang und Ende können wir eine zusätzliche Kante in unsere Tour einbauen, sodass unsere endgültige Länge $\frac{n-1}{2} \cdot 3 + 1$ beträgt. Für $n = 2027$ haben wir also eine Tour der Länge $\frac{2027-1}{2} \cdot 3 + 1 = 1013 \cdot 3 + 1 = 3040$.



LEDs

Aufgabenvorbereitung	Mark Neumann
Übersetzung (de)	Benjamin Schmid
Übersetzung (fr)	Cheng Zhong
Lösung Autor	Benjamin Schmid



Teil 1: Stoff ist Bühnenmanager und richtet eine Reihe von 10 LED-Scheinwerfern für ein Konzert ein. Derzeit leuchten einige Lichter gelb (EIN) und einige sind weiss (AUS).

Er möchte alle 10 Scheinwerfer auf EIN setzen, aber es gibt einen Haken: Das Bedienfeld ist seltsam verdrahtet. Er kann nur Tasten drücken, die zwei benachbarte Lichter gleichzeitig steuern. Wenn er eine Taste für zwei benachbarte Lichter drückt, wechseln beide ihren Zustand. (Wenn ein Licht EIN war, wird es AUS; wenn es AUS war, wird es EIN).

Du erhältst die Startsequenz der 10 Lichter wie im Bild gezeigt.

Wie viele Tastendrucke benötigt Stoff mindestens, um alle 10 Lichter auf EIN zu setzen?

- ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9

Teil 2: Stoff stellt fest, dass es aufgrund der seltsamen Verdrahtung des Bedienfelds einige Startmuster geben könnte, bei denen es unmöglich ist, alle 10 Lichter auf EIN zu setzen, egal wie viele Tasten er drückt, aber er ist sich nicht sicher.

Jedes der 10 Lichter kann anfangs entweder EIN oder AUS sein, was bedeutet, dass es viele verschiedene mögliche Startmuster gibt.

Von allen möglichen Startmustern für diese 10 Lichter: Wie viele können nach einigen (möglicherweise null) Operationen mit Stoffs Tasten für benachbarte Lichter erfolgreich vollständig auf EIN gesetzt werden?

.....



Teilaufgabe 1: Schalte die Lichter EIN

Wir müssen mindestens 6 Knöpfe drücken.

Wir können einige Beobachtungen machen:

- AUS, EIN umschalten ergibt EIN, AUS und umgekehrt.
- Einen Knopf zweimal zu drücken ergibt keinen Sinn, da der zweite Knopfdruck den ersten rückgängig macht. Dies gilt auch, wenn wir dazwischen andere Knöpfe gedrückt haben.
- Der letzte Knopfdruck muss auf AUS, AUS sein, sonst schalten wir immer ein Licht aus und sind nicht fertig.

Um die minimale Anzahl Knopfdrücke zu ermitteln, gehen wir nun einfach von links nach rechts vor. Wir müssen Licht 2 umschalten, wofür wir zwei Optionen haben: 1 und 2 umschalten, oder 2 und 3 umschalten. 1 und 2 umzuschalten funktioniert nicht, da dann Licht 1 AUS ist, aber es gibt keinen Knopf, den wir drücken können, um es wieder auf EIN zu setzen. Daher müssen wir 2 und 3 umschalten. Als Nächstes müssen wir Licht 3 wieder einschalten. Mit dem gleichen Argument gibt es nur eine verbleibende Option: 3 und 4 umschalten.

Wenn wir alle Lichter auf diese Weise durchgehen, können wir genau bestimmen, welche Knöpfe wir drücken müssen: 2 und 3, 3 und 4, 5 und 6, 6 und 7, 8 und 9, 9 und 10.

Teilaufgabe 2: Unmögliche Muster

Es gibt 512 Muster, bei denen wir erfolgreich alle Lichter auf EIN setzen können.

Insgesamt gibt es $2^{10} = 1024$ mögliche Muster. Ein Knopfdruck schaltet immer genau zwei Lichter um. Daher bleibt die Gesamtzahl der eingeschalteten Lichter entweder gleich, erhöht sich um zwei oder verringert sich um zwei. Insbesondere ändert sich die Parität der Gesamtzahl der Lichter nicht. Wenn wir also mit einer ungeraden Anzahl eingeschalteter Lichter beginnen, werden wir nie alle Lichter einschalten können. Wenn wir hingegen mit einer geraden Anzahl eingeschalteter Lichter beginnen, ist es immer möglich, alle einzuschalten, indem wir das Verfahren aus Teil 1 anwenden.

Mit dieser Beobachtung müssen wir die Anzahl der Muster mit einer geraden Anzahl eingeschalteter Lichter zählen. Das ist genau die Hälfte aller Muster: 512.



Cogros

Aufgabenvorbereitung	Mark Neumann
Übersetzung (de)	Benjamin Schmid
Übersetzung (fr)	Cheng Zhong
Lösung Autor	Benjamin Schmid

Teil 1: Im neuen Einkaufszentrum Cogros gibt es 256 automatische Türen. Sie sind alle derzeit AUS, und jede Tür hat ihren eigenen Schalter.

Du erhältst einen anonymen Hinweis, dass genau einer dieser 256 Schalter vollständig defekt ist, aber Du weisst nicht, welcher. Deine Aufgabe ist es, ihn zu finden!

Jede Minute kannst Du genau eine der folgenden beiden Aktionen ausführen:

- Einen Schalter umlegen: Wenn es der defekte Schalter ist, passiert nichts. Wenn es ein funktionierender Schalter ist, ändert er den Zustand der Tür (von AUS auf EIN oder von EIN auf AUS). Das Umlegen des Schalters erlaubt Dir jedoch nicht, die Tür zu sehen, sodass Du nicht sofort ihren neuen Zustand kennst.
- Eine Tür inspizieren: Du gehst zu einer bestimmten Tür, um sie anzusehen und festzustellen, ob sie gerade EIN oder AUS ist.

Wie viele Minuten benötigst Du mindestens, um garantiert den defekten Schalter zu finden, selbst im ungünstigsten Fall (wenn Du extrem viel Pech hast)?

☐ 384 ☐ 255 ☐ 256 ☐ 510 ☐ 512

Teil 2: Nachdem Du etwas länger darüber nachgedacht hast, stellst Du fest, dass es eine dritte Aktion gibt, die Du ausführen kannst. Genau wie die anderen dauert auch diese Aktion genau eine Minute:

- Den Stromzähler überprüfen: Du kannst den gesamten Stromverbrauch des gesamten Cogros-Einkaufszentrums messen. Dies sagt Dir sofort, wie viele Türen im gesamten Gebäude derzeit insgesamt EIN sind.

Mit dieser neuen Aktion zusammen mit den beiden ursprünglichen Aktionen: Wie viele Minuten benötigst Du nun mindestens, um garantiert den defekten Schalter zu finden?

Hinweis: Wenn Du mehrere Schalter umlegst, bevor Du eine Überprüfung durchführst, dauert jeder Schalter immer noch eine ganze Minute.

.....



Teilaufgabe 1: Finde die kaputte Tür

Wir brauchen mindestens 510 Minuten.

Es gibt keine Möglichkeit, mehrere Türen zu überprüfen oder mehrere Schalter gleichzeitig umzulegen. Daher dauert es genau zwei Minuten, um zu überprüfen, ob ein bestimmter Schalter kaputt ist: eine Minute, um den Schalter umzulegen, und eine Minute, um zu überprüfen, ob die Tür EIN ist.

Eine mögliche Strategie ist, einfach alle Schalter zu überprüfen, was $2 \cdot 256 = 512$ Minuten dauert. Es gibt eine kleine Optimierung: Sobald wir alle Schalter bis auf einen überprüft haben, müssen wir den letzten Schalter nicht mehr überprüfen, da wir wissen, dass genau einer kaputt ist. Das spart zwei Minuten, sodass 510 Minuten übrig bleiben. Dies ist die beste Lösung.

Teilaufgabe 2: Stromverbrauch

Wir brauchen mindestens 263 Minuten.

Die neue Operation ermöglicht es uns, mehrere Türen gleichzeitig zu überprüfen. Dies erlaubt eine Strategie namens binäre Suche. Zuerst legen wir genau die Hälfte der Schalter um. Dann lesen wir den Stromzähler ab. Wenn die Anzahl der eingeschalteten Türen mit der Anzahl der umgelegten Schalter übereinstimmt, wissen wir, dass alle diese Schalter korrekt funktioniert haben. Somit muss sich der kaputte Schalter in der anderen Hälfte befinden. Wenn die Anzahl der eingeschalteten Türen hingegen nicht übereinstimmt, wissen wir, dass einer der umgelegten Schalter kaputt war. Wir können daher die andere Hälfte ausschliessen.

Nach einem dieser Schritte konnten wir eine Hälfte der Schalter ausschliessen. Wir können dann dieselbe Strategie rekursiv auf die Hälfte anwenden, in der wir den kaputten Schalter vermuten. Sobald nur noch zwei mögliche Schalter übrig sind, überprüfen wir einen davon und haben den kaputten Schalter identifiziert.

Wir können die benötigte Anzahl Minuten dann mit der folgenden Tabelle berechnen.

Verbleibende Schalter	Zu überprüfende Schalter	Benötigte Minuten
256	128	129
128	64	65
64	32	33
32	16	17
16	8	9
8	4	5
4	2	3
2	1	2

Wenn wir die letzte Spalte aufsummieren, erhalten wir $129 + 65 + 33 + 17 + 9 + 5 + 3 + 2 = 263$ Minuten.