

# 2023-NZ-01 Brücken bauen!

## Body

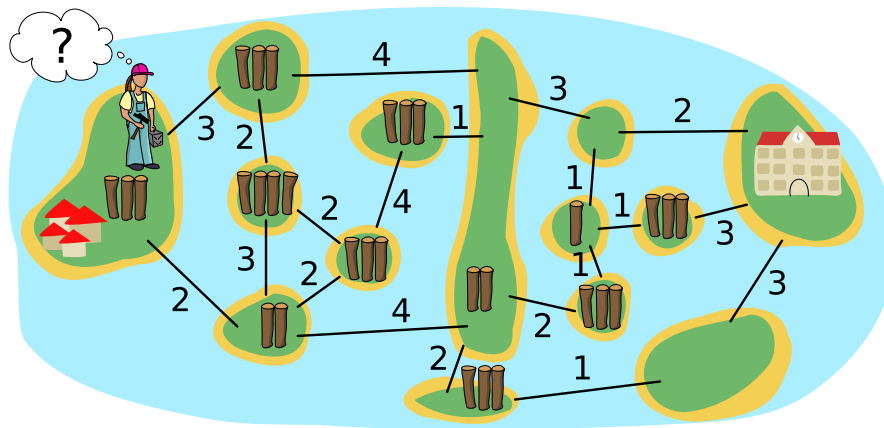
Auf der Insel ganz links sind Kinder eingezogen. Bianca soll Brücken bauen, über die die Kinder zur Schule auf der Insel ganz rechts gehen können.

Die Insel-Karte zeigt, wie viele Baumstämme es auf jeder Insel gibt. Diese Baumstämme kann Bianca nehmen, um an den Linien Brücken zu bauen. Die Zahl an einer Linie sagt, wie viele Baumstämme dort für eine Brücke benutzt werden. Sobald es zwischen zwei Inseln eine Brücke gibt, kann Bianca darüber gehen und Stämme, die sie noch hat, mitnehmen. Natürlich kann sie jeden Baumstamm nur für eine Brücke benutzen.

Bianca fängt auf der Insel links an. Ihr Ziel ist, möglichst wenige Baumstämme zu benutzen.

## Question/Challenge - for the brochures

*An welchen Linien soll Bianca Brücken bauen, damit sie ihr Ziel erreicht?*



## Interactivity instruction - for the online challenge

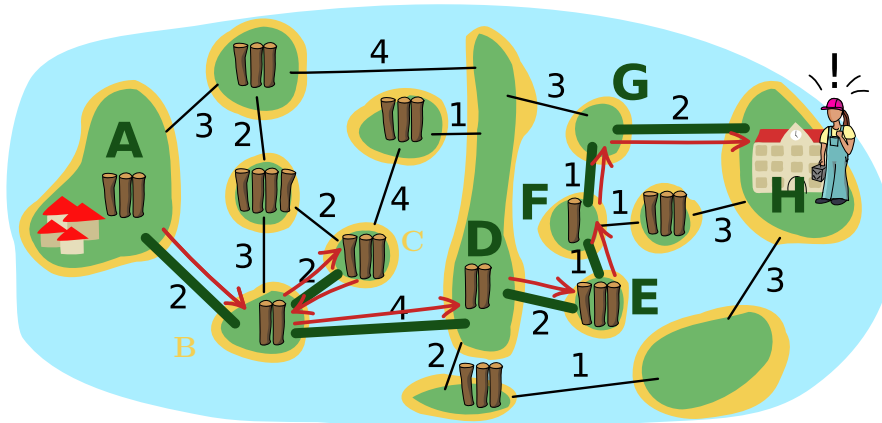
Klicke eine Linie an, um sie auszuwählen. In Biancas Denkblase siehst du, wie viele Baumstämme du benutzt. Wenn du fertig bist, klicke auf „Antwort speichern“!

## Answer Options/Interactivity Description

Each «path» must be clickable and when clicked, a bridge (maybe trunks in the appropriate number laid out?) appears. Any bridge could be built at any point in time, even impossible bridges (to allow the students to start from the school as well).

Es würde sich lohnen, die momentane Summe der angeklickten Brücken und die Summe der insgesamt zur Verfügung stehenden Baumstämme anzuzeigen; dann wäre jedoch das Problem, dass damit die feasibility noch nicht gezeigt ist, und dass dies zu Missverständnissen führen könnte.

So ist es richtig:



- Auf der Insel A nimmt sie die drei Baumstämme und benutzt zwei davon für die erste Brücke. Mit dem verbleibenden Baumstamm geht sie über die Brücke und hat auf der Insel B  $3 - 2 + 2 = 3$  Baumstämme. Das sind nicht genug, um eine Brücke zur Insel D zu bauen.
- Deshalb baut sie mit 2 Stämmen eine Brücke zur Insel C. Sie geht über die Brücke, nimmt die 3 Stämme von der Insel C und geht zurück. Nun hat sie  $3 - 2 + 3 = 4$  Stämme.
- Damit baut sie eine Brücke zur Insel D, geht über die Brücke und hat dann die 2 Stämme von Insel D.
- Damit baut sie eine Brücke zur Insel E und kann dort 3 Stämme nehmen. Sie baut weitere Brücken zu den Inseln F und G. Auf der Insel E hat sie also 3 Stämme, auf der Insel F  $3 - 1 + 1 = 3$  Stämme und auf der Insel G noch 2 Stämme.
- Die reichen genau, um eine Brücke zur Insel H mit der Schule zu bauen.

- Für die Brücken von Insel A bis Insel D hat Bianca 8 Stämme benutzt und kam ohne Stamm auf Insel D an. Wir notieren ihren Weg so:  $2 - [2, 2] - 4$  (von der Insel A über die Linie mit der 2 zur Insel B, dann zwischen B und C hin und zurück über die 2, dann über die 4 zu Insel D). Ein Weg mit weniger Stämmen wäre  $3 - 4$ , kann aber nur mit Umweg gebaut werden ( $3 - [2, 2] - 4$ ), verbraucht also 9 Stämme, wobei Bianca auf Insel D mit einem Stamm im Vorrat ankommt. Alle anderen Wege von Insel A bis D verbrauchen 9 Stämme oder mehr.
- Für die Brücken von Insel D bis H hat Bianca 6 Stämme benutzt. Den direkten Weg  $3 - 2$  kann sie nicht bauen, auch nicht mit einem Stamm im Vorrat. Alle anderen Wege von Insel D zu Insel H verbrauchen 6 Stämme oder mehr.

Es ist also nicht möglich, mit weniger als 14 Stämmen Brücken zu bauen, über die die Kinder von der Dorf-Insel A zur Schul-Insel H gehen können. Mit den von ihr gebauten Brücken hat Bianca also ihr Ziel erreicht.

## This is Informatics

Die Insel-Karte mit den durch Linien angezeigten «Brücken-Bauplätzen» kann als *Graph* modelliert werden: Das ist eine mathematische Struktur, die Objekte (auch Knoten genannt) paarweise miteinander in Relation setzt (die Paare nennt man auch Kanten). In einem Graphen man die Inseln als Knoten und die Linien als Kanten modellieren. Dabei haben die Kanten *Gewichte*, nämlich die Anzahl der für den Brückenbau entlang einer Linie benutzten Baumstämme, aber auch die Knoten (die Anzahl der Stämme auf einer Insel) – das ist eher ungewöhnlich. Für Graphen, bei denen nur die Kanten gewichtet sind, kennt die Informatik mehrere effiziente Algorithmen, die einen kürzesten Weg (über Kanten mit minimaler Summe der Gewichte) zwischen zwei Knoten berechnen können.

Das Problem, das Bianca in dieser Biberaufgabe optimal lösen möchte, ist komplizierter: Sie möchte zwar auch einen kürzesten Weg gehen, hat aber eine Randbedingung: Die Summe der Knotengewichte auf ihrem bisherigen Weg (die Stämme, die sie nehmen konnte) abzüglich der Summe der Kantengewichte auf ihrem Weg (die Stämme, die sie für den Brückenbau benutzt hat) muss größer sein als das Gewicht der Kante, die sie als nächste gehen bzw. wo sie eine Brücke bauen möchte. Um den optimalen Weg zu finden, müssen hier eventuell alle Möglichkeiten ausprobiert werden. Die Zerlegung des Problems in zwei Teile hilft, die Anzahl der Möglichkeiten zu reduzieren. Und wegen der Randbedingung kann man viele Möglichkeiten ausschließen, bevor man sie komplett probiert hat. In der Informatik ist ein solches Vorgehen (Probieren und Ausschließen) als *Backtracking* bekannt (siehe auch die Biberaufgabe «Gemüsebeet»).

## This is Computational Thinking

Beim Lösen des Problems begegnen wir einem vier wichtige Aspekte des Computational Thinking:

- Es ist im Grunde egal, dass es um Inseln und Brücken geht, alleine die Struktur des Graphen ist relevant. Diesen Vorgang nennt man Abstrahieren, wenn die wesentlichen Informationen verwendet werden und alle übrigen (die Einkleidung durch die Aufgabe) weggelassen werden.
- Das ziemlich komplexe Problem wird in zwei Teile zerlegt (manchmal auch Dekomposition genannt), die für sich genommen relativ einfach lösbar sind.
- Dadurch dass zu jeder Zeit genügend Baumstämme für den Bau von Brücken vorhanden sein müssen, ist auch algorithmisches Denken oder das Denken in Abläufen notwendig.
- Zuletzt wird in der Regel eine lokale Optimierung stattgefunden haben: man hat eine mögliche Lösung gefunden und versucht durch Ausprobieren von leicht veränderten Alternativen eine bessere Lösung zu finden.

Im Abschnitt «Dies ist Informatik» oben wird zudem Algorithmenanalyse aufgezeigt.

## Informatics Keywords and Websites

- Graph [https://de.wikipedia.org/wiki/Graph\\_\(Graphentheorie\)](https://de.wikipedia.org/wiki/Graph_(Graphentheorie))
- gewichteter Graph [https://de.wikipedia.org/wiki/Kantengewichteter\\_Graph](https://de.wikipedia.org/wiki/Kantengewichteter_Graph)

## Computational Thinking Keywords and Websites

- Abstraktion (<https://de.wikipedia.org/wiki/Abstraktion>)
- Dekomposition (<https://de.wikipedia.org/wiki/Modell#Modellbildung>)
- Algorithmisches Denken (<https://algdenken.phgr.ch>)
- Lokale Optimierung ([https://de.wikipedia.org/wiki/Lokale\\_Suche](https://de.wikipedia.org/wiki/Lokale_Suche))
- Algorithmenanalyse (<https://de.wikipedia.org/wiki/Algorithmus#Algorithmenanalyse>)