Sample Bebras Graph Question

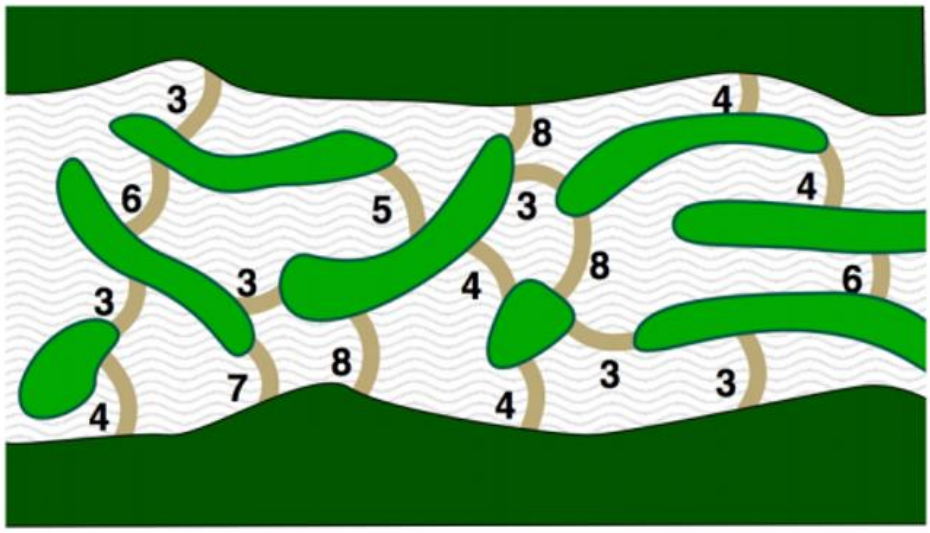
The international Bebras competition takes place each year in November. Free to enter, multiple choice tests are available for every age group. No prior practice is needed. The tests are automatically marked, and certificates awarded based on pupil achievement in relation to their school performance. An inclusive and motivational event, the two questions below give an idea of the range of questions. Booklets are available for download which include all questions, answers and their relation to Computer Science.

Beaver Logs

The beaver community is designing a new dam on the river. They want to use the least number of logs.

They are clever, so they want to take advantage of the small islands in the river.

The picture shows the river, the islands, and the number of logs needed to build each dam segment.



Question: What is the least number of logs needed for the new dam?

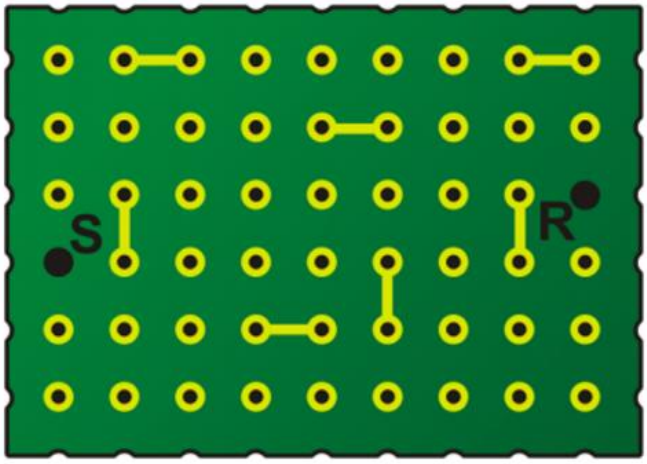
Target age: KS3

Building A Chip

A small chip is composed of a grid of contacts (marked as dots). Some are already connected (marked as line segments).

Connectors are always only between adjacent contacts, horizontally or vertically.

We want to connect S and R with a continuous sequence of connectors, which do not touch any already connected contacts.



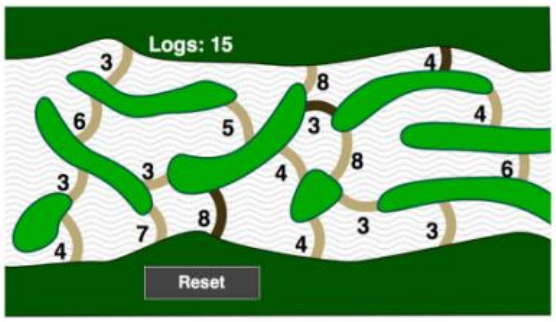
Question: How many different ways are there to connect S and R with the least possible number of connectors?

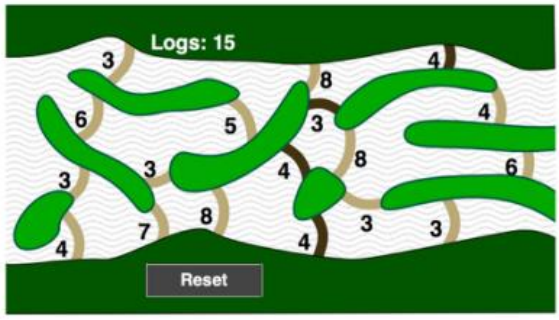
5, 13, 15 or 16

Target Age: Post 16

Answers and information

Beaver Logs





15 logs are needed, either answer shown right.

Explanation:

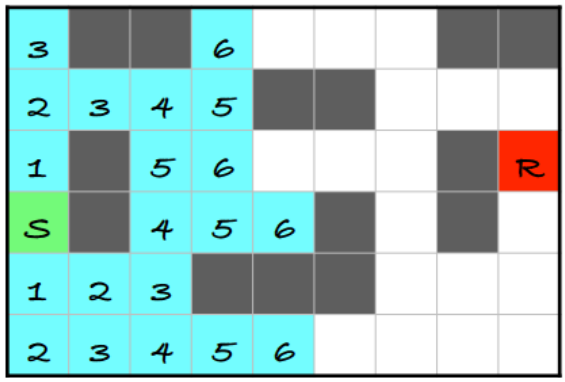
The easiest way to go about building this dam is to think of the problem as finding the shortest route across the river. The shortest route requires 15 logs but there is more than one way of achieving this. In the two answers above, the dam is built by connecting to the middle island. There are two ways of doing this from the bottom bank that both need 8 logs.

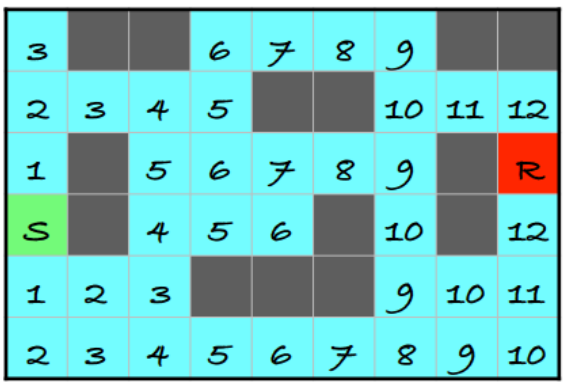
It's Computational Thinking:

Skills - Abstraction (AB), Algorithmic Thinking (AL), Decomposition (DE), Evaluation (EV), Generalisation (GE)

Computer scientists are lazy and smart, which is a great combination. They learn a bag of tricks, and whenever they encounter a problem, they try to apply one of them. In this case, they would observe that building a dam across the river is the same thing as getting to the other side with the least number of logs. In this way they change a new problem (building dams) into a well known one (finding the shortest path). The algorithm we used for solving it is called Dijkstra's algorithm. It was invented by E. W. Dijkstra, who was one of the most influential computer scientists and discovered many interesting algorithms.

Building A Chip





The correct answer is 15

Explanation:

It is not difficult to find some shortest paths. You can imagine a wave spreading over the board, one contact at a time, starting at S and moving toward R. When the wave reaches a contact, it takes the least possible number of connectors ("steps" of the wave).

The first table shows the wave midway through filling in. The numbers show the order in which the wave reached them, and the blacked out cells indicate the connectors we cannot connect to.

They are also the length of the shortest path to the respective contact. The highest numbers are the current edge of the wave. The table on the right shows the completion of the process.

But if you try to find all the shortest paths, you can easily get lost. Luckily, you do not really need to, you are just interested in the number of paths. So how can we break this task down in to some smaller tasks?

We need to realise this: A shortest path to any given contact must go through one of the adjacent contacts, which is exactly one connector closer to the start. If there are more such contacts, any of them can be used. So if you want to know the number of possibilities, you need to sum the possibilities to get to these adjacent contacts. The number of shortest paths to a certain contact is the sum of the numbers of the shortest paths to the adjacent contacts which are one connector closer to the start.

This allows you to follow a procedure, which will reliably give you the final number of shortest paths. You can proceed similarly as with finding the shortest path, filling in the table like a wave. Start at S. The contacts next to it can be reached in only one way. Then add the next contacts along the wave edge and always sum up the neighbouring numbers from the previous step.

The resulting table is shown below:

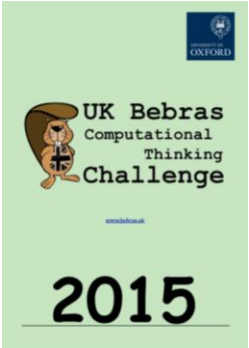
1			1	1	1	1		
1	1	1	1			6	6	6
1		2	4	5	5	5		15
S		1	1	1		8		9
1	1	1				3	6	9
1	2	3	3	3	3	3	3	3

It's Computational Thinking:
Skills - Abstraction (AB), Algorithmic Thinking (AL), Decomposition (DE), Evaluation (EV), Generalisation (GE)

Nowadays, integrated circuits (chips) are used in all electronic equipment and they have revolutionised the world of electronics. A chip can be made very compact, having up to several billion transistors and other components in an area the size of a fingernail. Without them all computation thinking would be theoretical.

The chips are very crowded and hence not easy to design. Computer scientists use many clever algorithms to do this. When deciding where to put two components, one criterion can be the number of paths between them: the more paths available, the less constraint for the rest of the board, because some third component will block these two components easily.

Finding the shortest path is a common problem in computational thinking. The described "wave" procedure is actually called a "breadth-first search". Here it was adapted to count all the shortest paths. We did not have to check all the possible paths, because we proceeded systematically from the start. This approach is called dynamic programming.



Bebras booklets of questions, answers and explanations for each annual competition can be downloaded from www.bebras.uk