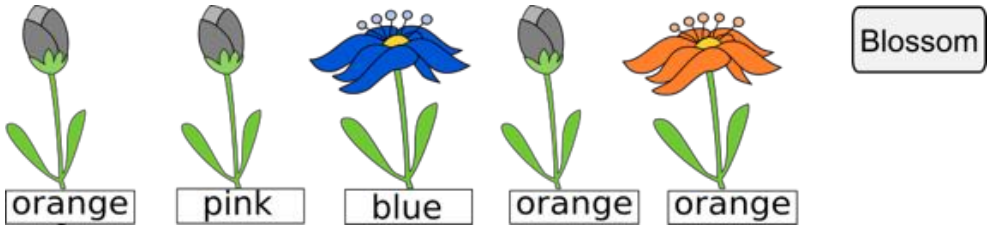
Jane is playing a computer game.

First the computer secretly chooses colours for five buds. The available colours for each flower are **blue**, **orange**, and **pink**. Jane has to guess which flower has which colour. She makes her first five guesses and presses the Blossom button.

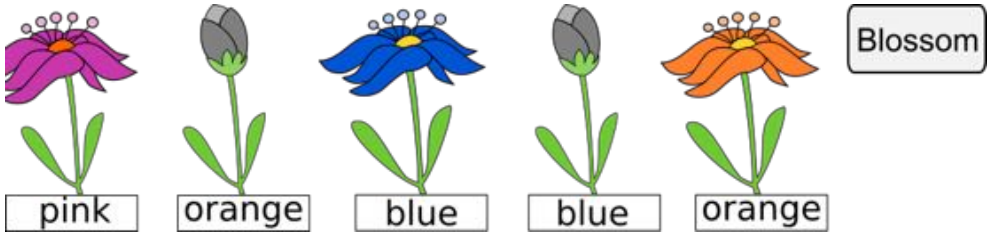
The buds, whose colours she guessed correctly, break into flowers. The others remain as buds.

Jane's first go:



Jane then has another go at guessing and presses the Blossom button again.

Jane's second go:



Question:

What colours did the computer choose for the flowers?

A. blue pink blue orange orange
B. pink blue blue blue orange
C. pink blue blue pink orange
D. pink pink blue pink orange

Answer:

C. pink blue blue pink orange

Explanation:

After two guesses, there are three blossomed flowers. So, we can already see the colour chosen by the computer for the first, third and fifth flower. The colour of the first flower is pink, so answer A) cannot be correct.

For the second flower Jane guessed pink in the first guess and it did not blossom, then she guessed orange and it did not blossom either. As there are only three colours available, the second flower must be blue. This rules out answer D).

Similarly, Jane chose orange and blue for the fourth flower and it still has not blossomed, so it must be pink. And this rules out answer B)

Answer C) must therefore be correct.

It's Computational Thinking:

Abstraction
Data representation - symbolism
Data interpretation – patterns
Algorithms - following

Drawing consequences from events that happened or did not happen is an important ability for solving many kinds of problems. The task is a simplified version of the Mastermind board game. It is simplified because after each guess the player gets complete information about all the flowers. If in each guess the player chooses a different colour for the not-yet-blossomed flowers, then in the third guess he/she can always correctly pick the colours of all the flowers.

Magic Potions

Years 3+4		Years 7+8	B
Years 5+ 6	C	Years 9+10	A
		Years 11+12	A

Betaro Beaver has discovered five new magic potions:

one makes ears longer
another makes teeth longer
another makes whiskers curly
another turns the nose white
the last one turns eyes white.

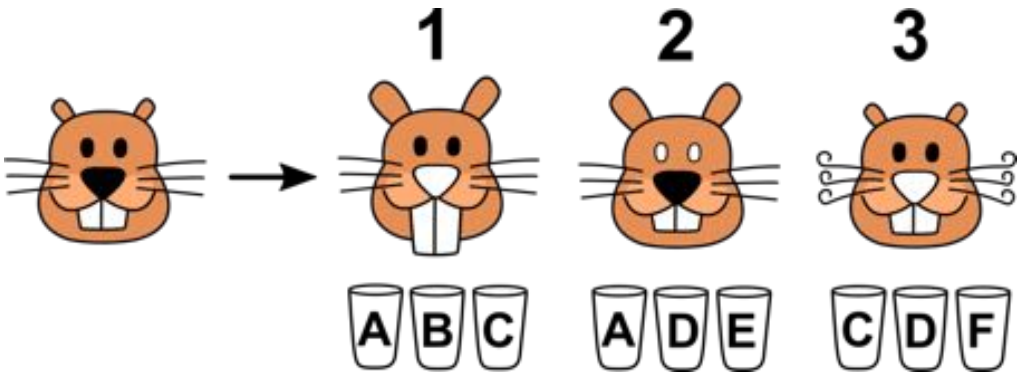
Betaro put each magic potion into a separate beaker. He put pure water into another beaker, so there are six beakers in total. The beakers are labeled A to F. The problem is, he forgot to record which beaker contains which magic potion!

To find out which potion is in each beaker, Betaro set up the following experiments:

Expt 1: A beaver drinks from beakers A, B and C together - the effects are shown in Figure 1.

Expt 2: A beaver drinks from beakers A, D and E together - the effects are shown in Figure 2.

Expt 3: A beaver drinks from beakers C, D and F together - the effects are shown in Figure 3.



Question:
Which beaker contains pure water?

Answer:

Beaker D

Explanation:

Solution 1:
By Experiment 1, none of A, B and C is pure water, since there are three changes that happen to the beaver.
By Experiment 2, either D or E is pure water or the magic potion making his nose white since A is not pure water, from Experiment 1.
By Experiment 3, D and F are pure water or the magic potion making his whiskers curly, since C is not pure water, again from Experiment 1.
Therefore, D is pure water.

Solution 2:
Experiment 1 has three effects, Experiment 2 and 3 both have two effects. Therefore, there is no pure water in Experiment 1 and there is exactly one water beaker in Experiment 2 and Experiment 3. The only common beaker between experiments 2 and 3 is beaker D. Thus, D is pure water.

It's Computational Thinking:

Algorithms – following and describing
Digital systems

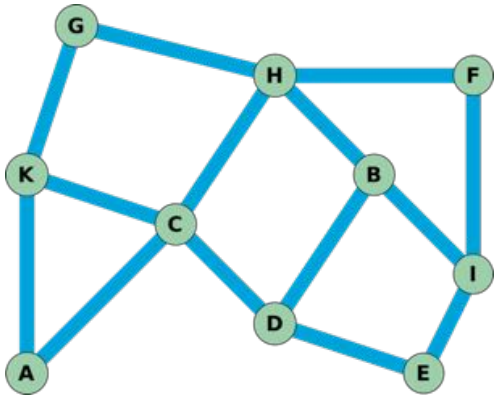
In this problem, we have a collection of facts that we need to find new information from. This can be done using logical reasoning. Logic plays an important role in Computer Science. The smallest unit a computer works with is a bit, which has a value of 1 (true) or 0 (false). All other information in a computer is stored using a specific combination of bits. The computer uses logic to figure out what decisions it should follow, and each of these decisions is based on whether certain bits are set to true (1) or false (0).

This problem also explores basic set theory. We are looking for an element in the set which is not in the set used in Experiment 1, which means it is the complement of A, B, C. We then look at the intersection of Experiments 2 and 3 in order to determine the common element in both.

Doctor Hamid wants to build three hospitals for the beavers.
The hospitals can only be built on the places shown on the map below.
To get to a hospital, the beavers should not have to swim through more than one stream from any of these places.

Question:

Choose three places to build the hospitals for Doctor Hamid.



Answer:

There are several correct solutions, one for instance uses the places E, H and K:

- For the places D, E and I the beavers can swim to E.
- For the places B, C, F, G and H the beavers can swim to H.
- For the places A, C, G and K the beavers can swim to K.

The other solutions are: A E H, C G I, C H I, C I K, D F K, B I K and C E H.

Explanation:

The solutions can be found by placing a station at a random position and marking all stations that are reachable within one step. Then you can position the next station and so on. Once all three stations are placed there are two possibilities: either it's a solution or there are one or more places that are not marked. If it's not a solution, you can remove the last station you've placed and place it in another place and check again. If you are still not lucky to find a solution with 3 stations you have to "backtrack" and place the last station on another place. By doing this systematically one can find all possible solutions.

It's Computational Thinking:

Abstraction
Data interpretation – patterns and models
Impacts - sustainablity

In Computing this channel system is generalised to the concept of a graph consisting of vertices (intersections) and edges (water canals). The more general problem is to find a so-called "vertex cover" in a graph. This is a subset of the vertices that cover the whole graph. Whenever all the neighbour vertices to this subset are added together, they will cover all vertices of the graph. Usually a minimal number of such vertices is the most cost efficient. In more complex graphs it is very hard to find these cost-effective subsets of the vertices. It needs a computer algorithm to find a solution.

The method of placing the stations described in the explanation section is called backtracking. You try one solution and if it is not correct you take back the last step you've made and try another step, ideally systematically until you have exhausted all possible last steps. Then you take back the pre-last step, try all solutions and so on until you have found a correct solution. This method is not very efficient, but for this kind of problems it works reliably.

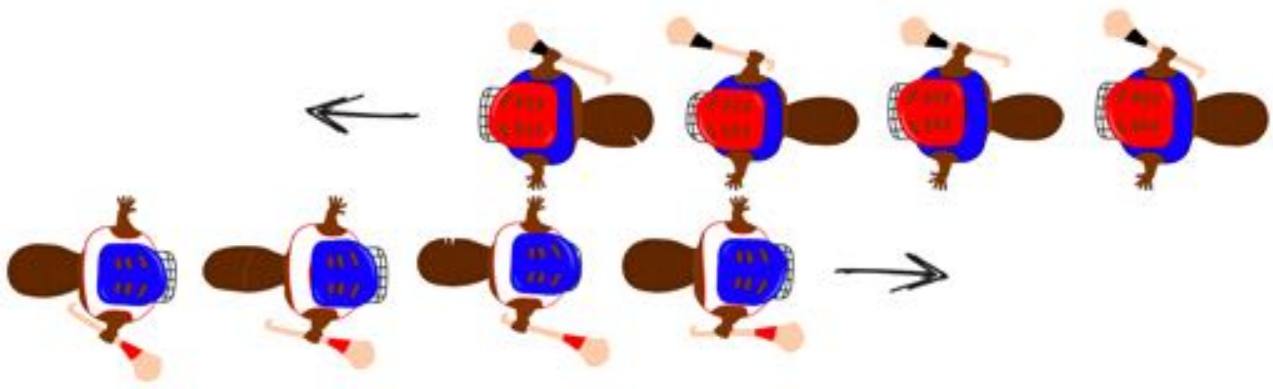
Hurlers Shake Hands

Years 3+4
Years 5+ 6 C
Years 7+8 B
Years 9+10 A
Years 11+12

Beavers enjoy playing hurling, a popular game similar to hockey.

After the game ends, the beavers in each of the two teams line up in a row and walk past the other team. As they pass each other, they shake hands. At the beginning, only the first player on each team shakes hands. Next, the first two players shake hands (see picture below). This continues until each player has shaken hands with every player on the other team.

There are 15 players on each team.



Question:

If each player takes one second to shake hands and move to the next player, how many seconds of shaking hands will there be?

Answer:

29

Explanation:

The amount of handshaking is exactly the length of one line plus the length of the other line, minus one.

Let us imagine that there is only 1 player on each team. After 1 second, all handshaking has finished. Let us imagine that there are only 2 players on each team. During the first second, the first player on each team shakes hands. During the second second, the first player on each team is shaking hands with the second player on the other team, and during the third second, the second two players are shaking hands with each other. So, that's three seconds.

With 15 players in each team, the number of seconds required is 15 + 15 – 1 = 29.

It's Computational Thinking:

Algorithms – following and describing
Digital systems
Specification – techniques
Interactions – data and processes
Impacts - sustainability

This task can be viewed as an illustration of a parallel processing paradigm called pipeline processing. Pipeline processing is a very efficient way to get many computers working together to solve problems quickly, but it can take a relatively long time to reach that efficiency, just like our players at the back of each line had to wait quite a while before their first handshake.

Analysing the running time of an algorithm is a sophisticated part of computer science called computational complexity analysis. In this Task, we know the team size is fixed at 15, and can deduce that the "running time" of the hand shaking algorithm is 29 seconds. However, in computational complexity we would be asked to measure the running time independent of a specific team size. We would conclude that the hand shaking algorithm takes 2N-1 seconds, for any team size N, where N is 1, or any larger natural number.

Concurrent Directions

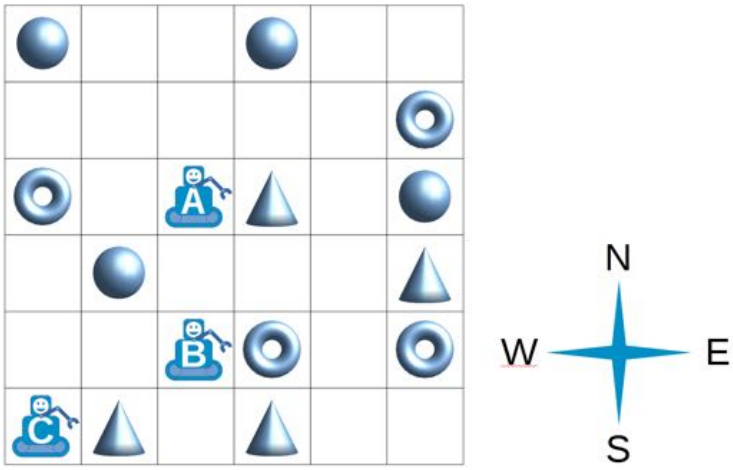
Years 3+4
Years 5+ 6 C
Years 7+8 B
Years 9+10 A
Years 11+12

In a warehouse, three robots always work as a team.

When the team gets a direction instruction (N, S, E, W), all robots in the grid will move one square in that direction at the same time.

After following a list of instructions, the robots all pick up the object found in their final square.

For example, if we give the list N, N, S, S, E to the team, then robot A will pick up a cone, robot B will pick up a ring, and robot C will pick up a cone.



Question:
Which list of instructions can be sent to the robots so that the team picks up exactly a sphere, a cone, and a ring?

A. N, E, E, E
B. N, E, E, S, E
C. N, N, S, E, N
D. N, E, E, S, W

Answer:

B. N, E, E, S, E

Explanation:

A. If the team's list is N, E, E, E then robot A will pick up a ring, robot B will pick up a cone, and robot C will pick up a ring. No sphere is picked up, so this is an incorrect answer.

B. If the team's list is N, E, E, S, E then robot A will pick up a sphere, robot B will pick up a ring, and robot C will pick up a cone. There is one of each type of object, so this is the correct answer.

C. If the team's list is N, N, S, E, N then robot A will pick up a sphere, robot B will pick up a cone, and robot C will pick up a sphere. No ring is picked up, so this is an incorrect answer.

D. If the team's list is N, E, E, S, W then robot A will pick up a cone, robot B will pick up a ring, and robot C will pick up a cone. No sphere is picked up, so this is an incorrect answer.

It's Computational Thinking:

Algorithms – following
Digital systems
Specification – techniques
Interactions – data and processes
Impacts – sustainability

When robots or computers work together at the same time, we say that they are working in parallel.

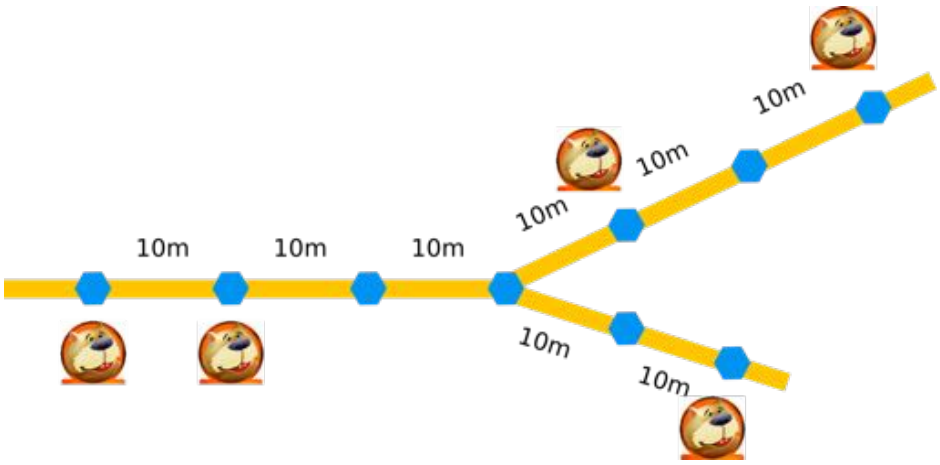
If we have a small number of robots or computers working in parallel, we could give each of them a different list of instructions. However, if we have thousands or millions of computers working together then it is not practical to write separate instructions for each. We have to give large groups of them the exact same instructions. For example, the Tianhe-2 supercomputer has 3 million separate computing cores that can be used to work together on solving a single complicated problem.

In this task, we have the potential for an additional complication. The robots are working in the same space, and their instructions must be prepared more carefully to ensure that they do not crash into each other or otherwise block one another. Preparing parallel instructions in such a situation is an extremely challenging aspect of computer science called concurrent programming.

Bus Stop

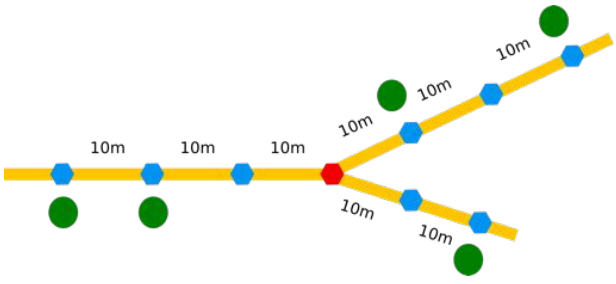
Years 3+4
Years 5+ 6 C
Years 7+8 A
Years 9+10
Years 11+12

The lodges of five beavers are shown on the map below.
The Beavers want to put a bus stop in one of the places marked by blue hexagons.
All the hexagons are 10m apart.
The beavers decide that the sum of the distances from their lodges to the bus stop must be as small as possible.



Question:
Click on the best place for the bus stop.

Answer:



Explanation:

One naive way to see this answer is to try all 9 possible locations for the bus stop and calculate the sum of the distances. This requires a lot of calculation. To reduce the amount of calculation, we can create a model or formula for our solution.

Suppose we located the bus stop to be at the branching of the roads (where all three roads intersect). The sum of all the distances from the lodges to the bus stop:

$30 + 20 + 10 + 30 + 20 = 110$

When the bus stop is moved x meters to the left, the first two distances will be reduced by x metres, and the distance from the last three lodges to the bus stop the will increase by x metres:

$(30-x) + (20-x) + (x + 10) + (30 + x) + (x + 20) = 110 + x$

We will get the same result if we choose a location for the bus stop at a distance of x meters to the right on the upper right route:

$(30 + x) + (20 + x) + (x + 10) + (30 - x) + (-x + 20) = 110 + x$

If you choose the location for the bus stop at a distance of x meters to the right on the lower right route, the result will be:

$(30 + x) + (x + 20) + (10 + x) + (x + 30) + (20 - x) = 110 + 3x$

It's Computational Thinking:

Data representation - symbolism
Breaking down problems into parts (Decomposition)
Data interpretation – patterns and models
Algorithms – following and describing
Impacts – sustainability

This is a classic problem of tree centre finding. It is also an optimisation problem. In computer science, many problems involve finding the best, or least cost, or minimal or maximal value to some function in order to save the most money, use the least amount of resources, take the least amount of time, etc. This problem is a simple case of a real-world problem involving transportation networks: City planners need to make these sorts of decisions for subway, bus or train locations in order to optimise the use of those systems. In computer science, we apply various algorithms to help find the optimal answer. For this problem, we can use one of three possible algorithms:

1) An exhaustive search of all options (for a small number of vertices). For larger networks (if there were thousands of locations) this is not feasible.

2) If the locations formed one row, we can simply compute the median location, and it will be optimal.

3) The optimal node has the property that, if we looked down each path from each of its neighbours, there is less than the half of the beavers living there. To optimise this criterion, we can use a dynamic-programming solution to find this answer in an efficient time.

Cards and Cones

Years 3+4
Years 5+ 6

Years 7+8
Years 9+10 B
Years 11+12 A

Inés has a pack of cards; each card has a number written on it from 1 to 9. The pack contains many of the same cards.

She places three coloured cones in front of her:

Inés intends to create stacks under the cones with the numbers facing up.
Each time she puts a new card on the stack it will cover the rest of the stack.

Her friend, Jules, takes notes as Inés puts cards, one at a time, under the cones:

Inés starts by placing a card with the number 5 on it under the red cone. Jules writes: **A <-- 5**
Next Inés places another card under the red cone, on top of the previous one. Jules writes: **A <-- 3**
Then Inés peeps under the red cone and finds a card from the pack that has the same number as she sees. She places it under the blue cone. Jules writes **B <-- A**

Jules' final notes look like this:

A <-- 5
A <-- 3
B <-- A
B <-- 3
A <-- B
B <-- 5
A <-- 6
C <-- B
A <-- B
B <-- 1

Question:
What cards are visible when the cones are lifted?
Write the correct numbers in the spaces below the question mark.