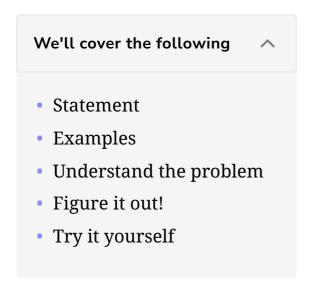
Matchsticks to Square

Try to solve the Matchsticks to Square problem.



Statement

Given an integer array, matchsticks, where matchsticks[i] is the length of the ith matchstick. Use every single matchstick to create a square. No stick should be broken, although they can be connected, and each matchstick can only be used once.

Return TRUE if we can make this square and FALSE otherwise.

Constraints:

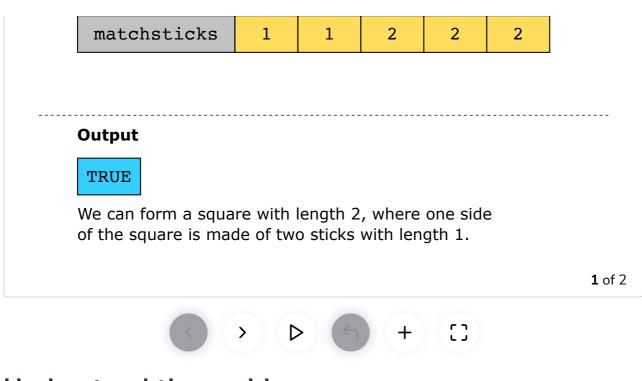
- $1 \leq \mathtt{matchsticks.length} \leq 15$
- $1 \leq \mathsf{matchsticks[i]} \leq 10^8$

Examples



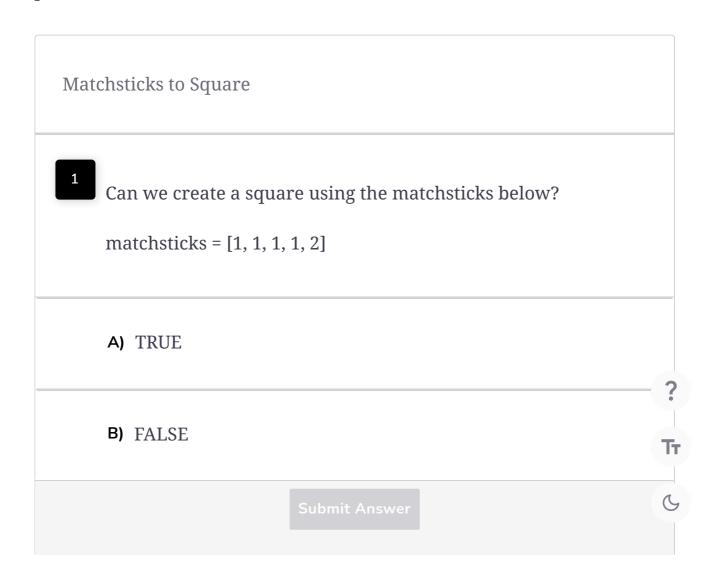
Input

6



Understand the problem

Let's take a moment to make sure you've correctly understood the problem. The quiz below helps you check if you're solving the correct problem:



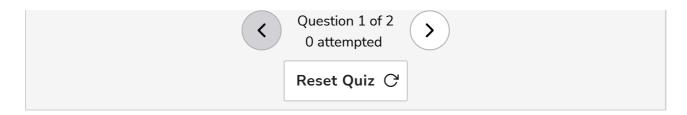
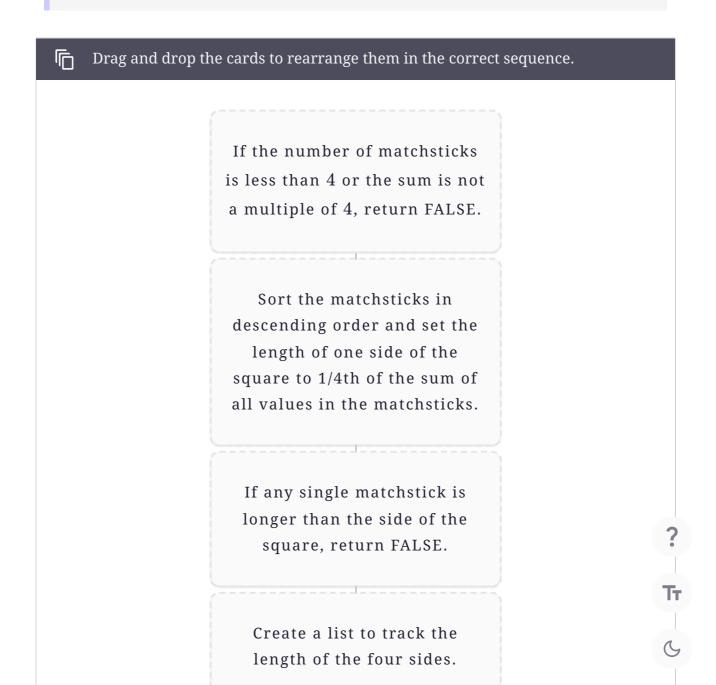


Figure it out!

We have a game for you to play. Rearrange the logical building blocks to develop a clearer understanding of how to solve this problem.

Note: As an additional challenge, we have intentionally hidden the solution to this puzzle.



Initialize it to [0, 0, 0, 0].

Define a backtracking function with the following base case: If all matchsticks are used, and the four sides are of equal length, return TRUE

Iterate through the list of four sides. For each side, if the sum of matchsticks[index] and the current side length is less than or equal to the target side length, update the current side by adding matchsticks[index] and recursively call the backtracking function with the next index.

If the backtracking function returns FALSE, undo the previous addition to the side length by subtracting the matchsticks[index] and move on checking if the current matchstick can help build the next side of the square.

Otherwise, return TRUE.

Return FALSE if no combination of matchsticks results in a valid square after

Īτ

9

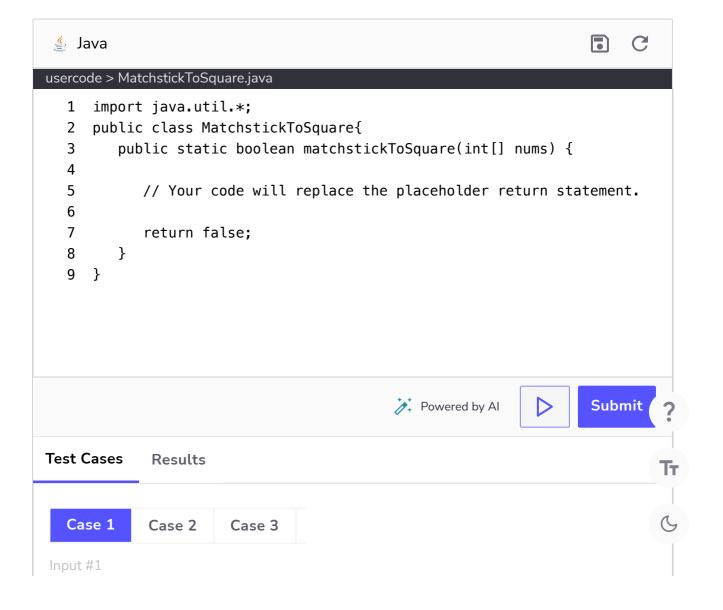


Try it yourself





Note: We have left the solution to this challenge as an exercise for you. You may try to translate the logic of the solved puzzle into a coded solution.



Matchsticks to Square



Sudoku Solver

Next →

Dynamic Programmin...



?



