

Matchsticks to Square

Try to solve the Matchsticks to Square problem.

We'll cover the following ^

- Statement
- Examples
- Understand the problem
- Figure it out!
- Try it yourself

Statement

Given an integer array, `matchsticks`, where `matchsticks[i]` is the length of the *i*th 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:

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

Examples

Sample example 1

Input

?

Tr



matchsticks	1	1	2	2	2
-------------	---	---	---	---	---

Output

TRUE

We can form a square with length 2, where one side of the square is made of two sticks with length 1.

1 of 2



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:

Matchsticks to Square

1

Can we create a square using the matchsticks below?

matchsticks = [1, 1, 1, 1, 2]

A) TRUE

B) FALSE

Submit Answer





Question 1 of 2
0 attempted



Reset Quiz ↺

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.



Drag and drop the cards to rearrange them in the correct sequence.

If the number of matchsticks is less than 4 or the sum is not a multiple of 4, return FALSE.

Sort the matchsticks in descending order and set the length of one side of the square to 1/4th of the sum of all values in the matchsticks.

If any single matchstick is longer than the side of the square, return FALSE.

Create a list to track the length of the four sides.



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



trying all possible combinations.

Reset

Show Solution


Submit

Try it yourself

Implement a solution in the following editor environment.



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.

 Java



usercode > MatchstickToSquare.java

```
1 import java.util.*;
2 public class MatchstickToSquare{
3     public static boolean matchstickToSquare(int[] nums) {
4
5         // Your code will replace the placeholder return statement.
6
7         return false;
8     }
9 }
```

 Powered by AI



Submit



Test Cases

Results

Case 1

Case 2

Case 3

Input #1

Tr



[1,1,2,2,2]

Matchsticks to Square

← Back

Sudoku Solver

Next →

Dynamic Programmin...



Mark as
Completed

