Jump Game I

Try to solve the Jump Game I problem.



Statement

In a single-player jump game, the player starts at one end of a series of squares, with the goal of reaching the last square.

At each turn, the player can take up to s steps towards the last square, where s is the value of the current square.

For example, if the value of the current square is 3, the player can take either 3 steps, or 2 steps, or 1 step in the direction of the last square. The player cannot move in the opposite direction, that is, away from the last square.

You have been tasked with writing a function to validate whether a player can win a given game or not.

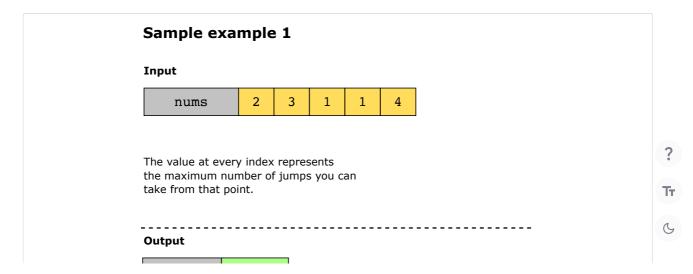
You've been provided with the nums integer array, representing the series of squares. The player starts at the first index and, following the rules of the game, tries to reach the last index.

If the player can reach the last index, your function returns TRUE; otherwise, it returns FALSE.

Constraints:

- $1 \leq \mathsf{nums.length} \leq 10^3$
- $0 \le \text{nums[i]} \le 10^3$

Examples





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:

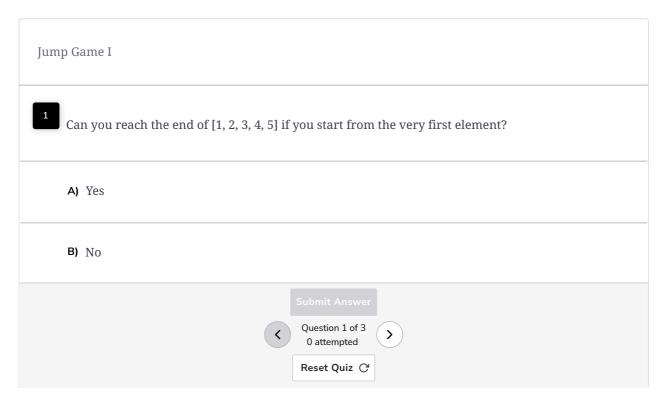
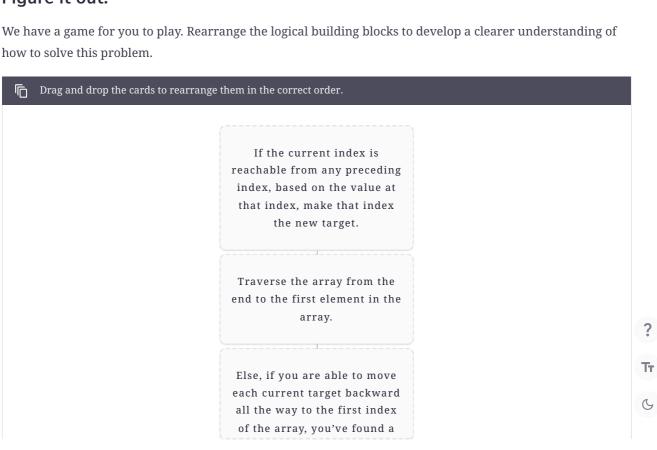
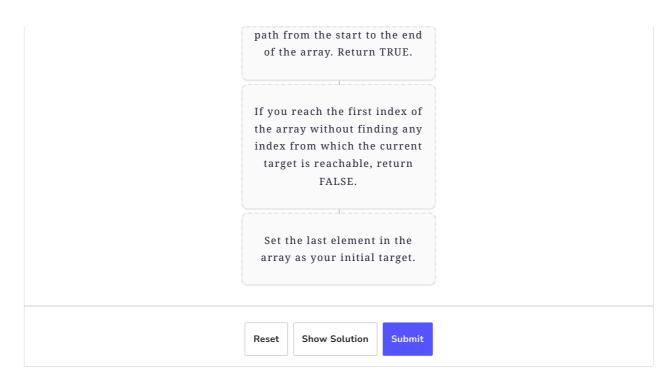


Figure it out!





Try it yourself

Greedy Techniques: In...

 \equiv

Implement your solution in the following coding playground

<u>≘</u> Java usercode > JumpGame.java 1 public class JumpGame{ public static boolean jumpGame(int[] nums) { // Your code will replace the placeholder return statement. 4 5 return false; 7 } Submit Powered by Al Test Cases Results Case 1 Case 2 Case 3 Input #1 [2,3,1,1,9] Jump Game I ← Back Next -

Tτ

Solution: Jump Game I

✓ Mark as Completed

?

Тт

C