

Practice 02 (08-03-2021)**Recursion II (Relatively Difficult Problems)**

Write recursive function for the following problems: Sample runs are given with each question.

1. Given an array of integers, is it possible to choose a group of some of the integers, such that the group sums to the given target? This is a classic backtracking recursion problem. Once you understand the recursive backtracking strategy in this problem, you can use the same pattern for many problems to search a space of choices. Rather than looking at the whole array, our convention is to consider the part of the array starting at index start and continuing to the end of the array. The caller can specify the whole array simply by passing start as 0. No loops are needed -- the recursive calls progress down the array.

groupSum(0, [2, 4, 8], 10)	TRUE
groupSum(0, [2, 4, 8], 14)	TRUE
groupSum(0, [2, 4, 8], 9)	FALSE
groupSum(0, [2, 4, 8], 8)	TRUE
groupSum(1, [2, 4, 8], 8)	TRUE
groupSum(1, [2, 4, 8], 2)	FALSE
groupSum(0, [1], 1)	TRUE
groupSum(0, [9], 1)	FALSE
groupSum(1, [9], 0)	TRUE
groupSum(0, [], 0)	TRUE
groupSum(0, [10, 2, 2, 5], 17)	TRUE
groupSum(0, [10, 2, 2, 5], 15)	TRUE
groupSum(0, [10, 2, 2, 5], 9)	TRUE

2. Given an array of integers, is it possible to choose a group of some of the integers, beginning at the start index, such that the group sums to the given target? However, with the additional constraint that all 6's must be chosen. (No loops needed.)

groupSum6(0, [5, 6, 2], 8)	TRUE
groupSum6(0, [5, 6, 2], 9)	FALSE
groupSum6(0, [5, 6, 2], 7)	FALSE
groupSum6(0, [1], 1)	TRUE
groupSum6(0, [9], 1)	FALSE
groupSum6(0, [], 0)	TRUE
groupSum6(0, [3, 2, 4, 6], 8)	TRUE
groupSum6(0, [6, 2, 4, 3], 8)	TRUE
groupSum6(0, [5, 2, 4, 6], 9)	TRUE
groupSum6(0, [6, 2, 4, 5], 9)	FALSE
groupSum6(0, [3, 2, 4, 6], 3)	TRUE
groupSum6(0, [1, 6, 2, 6, 4], 12)	TRUE
groupSum6(0, [1, 6, 2, 6, 4], 13)	TRUE
groupSum6(0, [1, 6, 2, 6, 4], 4)	FALSE
groupSum6(0, [1, 6, 2, 6, 4], 9)	TRUE
groupSum6(0, [1, 6, 2, 6, 5], 14)	TRUE
groupSum6(0, [1, 6, 2, 6, 5], 15)	TRUE
groupSum6(0, [1, 6, 2, 6, 5], 16)	FALSE

3. Given an array of integers, is it possible to divide the integers into two groups, so that the sum of the two groups is the same, with these constraints: all the values that are multiple of 5 must be in one group, and all the values that are a multiple of 3 (and not a multiple of 5) must be in the other. (No loops needed.)

plit53([1, 1])	TRUE
split53([1, 1, 1])	TRUE
split53([2, 4, 2])	TRUE
split53([2, 2, 2, 1])	TRUE
split53([3, 3, 5, 1])	TRUE
split53([3, 5, 8])	TRUE
split53([2, 4, 6])	TRUE
split53([3, 5, 6, 10, 3, 3])	TRUE

4. Given an array of integers, is it possible to divide the integers into two groups, so that the sums of the two groups are the same. Every integer must be in one group or the other. Write a recursive helper method that takes whatever arguments you like, and make the initial call to your recursive helper from splitArray(). (No loops needed.)

splitArray([2, 2])	TRUE
splitArray([2, 3])	TRUE
splitArray([5, 2, 3])	TRUE
splitArray([5, 2, 2])	TRUE
splitArray([1, 1, 1, 1, 1, 1])	TRUE
splitArray([1, 1, 1, 1, 1])	TRUE
splitArray([])	TRUE
splitArray([1])	TRUE
splitArray([3, 5])	TRUE
splitArray([5, 3, 2])	TRUE
splitArray([2, 2, 10, 10, 1, 1])	TRUE
splitArray([1, 2, 2, 10, 10, 1, 1])	TRUE
splitArray([1, 2, 3, 10, 10, 1, 1])	TRUE