

# 473. Matchsticks to Square

Add to List

DescriptionHintsSubmissionsSolutions

- Total Accepted: **7398**
- Total Submissions: **21515**
- Difficulty: **Medium**
- Contributors:weezer

Remember the story of Little Match Girl? By now, you know exactly what matchsticks the little match girl has, please find out a way you can make one square by using up all those matchsticks. You should not break any stick, but you can link them up, and each matchstick must be used **exactly** one time.

Your input will be several matchsticks the girl has, represented with their stick length. Your output will either be true or false, to represent whether you could make one square using all the matchsticks the little match girl has.

**Example 1:**

**Input:** [1,1,2,2,2]**Output:** true

**Explanation:** You can form a square with length 2, one side of the square came two sticks with length 1.

**Example 2:**

**Input:** [3,3,3,3,4]**Output:** false

**Explanation:** You cannot find a way to form a square with all the matchsticks.

**Note:**

1. The length sum of the given matchsticks is in the range of 0 to  $10^9$ .

2. The length of the given matchstick array will not exceed 15.

[Subscribe](#) to see which companies asked this question.

```
class Solution {
public:

    bool dfs(vector<int>& nums,vector<int> &slidelength, int index, int
target)
    {
        if(index==nums.size())
        {
            return          slidelength[0]==slidelength[1]          &&
slidelength[1]==slidelength[2] && slidelength[2]==slidelength[3];
        }
        for(int i=0;i<4;++i)
        {
            if(slidelength[i]+nums[index]>target) continue;
            slidelength[i]+=nums[index];
            if(dfs(nums,slidelength,index+1,target)) return true;
            slidelength[i]-=nums[index];
        }
        return false;
    }

    bool makesquare(vector<int>& nums) {
        if(nums.size()<4) return false;
        int sum = 0;
        for(auto k: nums) sum+=k;
        if(sum%4!=0) return false;
        int target = sum/4;
        vector<int> slidelength(4,0);
        return dfs(nums,slidelength,0,target);
    }
};
```