

还记得童话《卖火柴的小女孩》吗？现在，你知道小女孩有多少根火柴，请找出一种能使用所有火柴拼成一个正方形的方法。不能折断火柴，可以把火柴连接起来，并且每根火柴都要用到。

输入为小女孩拥有火柴的数目，每根火柴用其长度表示。输出即为是否能用所有的火柴拼成正方形。

示例 1:

输入: [1,1,2,2,2]

输出: true

解释: 能拼成一个边长为2的正方形，每边两根火柴。

示例 2:

输入: [3,3,3,3,4]

输出: false

解释: 不能用所有火柴拼成一个正方形。

思路，将火柴分为4个边，进行遍历，若每根火柴放在一条边上，能放就进行下一个，看看最终能否拼成正方形。

```
class Solution {
public:
    bool makesquare(vector<int>& nums) {
        if(nums.empty() || nums.size() < 4) return false;
        int sum = accumulate(nums.begin(), nums.end(), 0);
        if(sum % 4 != 0) return false;
        vector<int> sums(4, 0); //长度为4的数组sums来保存每个边的长度和
        sort(nums.rbegin(), nums.rend());
        return helper(nums, sums, 0, sum/4);
    }
    bool helper(vector<int>& nums, vector<int>& sums, int pos, int target)
    {
        if(pos >= nums.size()){
            return sums[0] == target && sums[1] == target && sums[2] ==
target && sums[3] == target;
        }
        //对于当前这个火柴，尝试拼入上下左右四个边
        for(int i=0; i<4; ++i){
            if(sums[i]+nums[pos] > target) continue;
            sums[i] += nums[pos]; //把当前火柴从i个边中拿出来，好尝试下一条边
            if(helper(nums, sums, pos+1, target)) return true; //如果这个火
柴被成功使用，就开始尝试拼下一根火柴
            sums[i] -= nums[pos]; //用当前火柴拼第i个边
        }
    }
}
```

```
        return false;
    }
};
```