zhuanlan.zhihu.com

leetcode No.698 划分为k个相等的子集

5-6 minutes

题目链接:

题目描述:

给定一个整数数组 nums 和一个正整数 k，找出是否有可能把这个数组分成 k 个非空子集，其总和都相等。

示例 1:

```
输入： nums = [4, 3, 2, 3, 5, 2, 1], k = 4
输出： True
说明： 有可能将其分成 4 个子集（5），（1,4），（2,3），（2,3）等于总和。
```

注意:

- 1 <= k <= len(nums) <= 16
- 0 < nums[i] < 10000

解题思路:

其实说白了就是要凑数，凑出 k 个 sum(nums)/k 就可以

把这个数组分成 k 个非空子集，其总和都相等。

如果问题转换成了凑数，我们就能反应过来要用回溯法了，写回溯法一般都是套路，不熟悉就多做几题练练，本题在具体实现的时候要注意返回条件和元素使用与否。

代码如下：

```
class Solution {
public:
    bool canPartitionKSubsets(vector<int>& nums, int k) {
        int n = nums.size();
        int sum = accumulate(nums.begin(), nums.end(), 0);
        if(sum % k != 0) return false;
        int target = sum / k;
        vector<int> used(n, 0);
        return backtrack(nums, k, target, 0, 0, used);
    }
    bool backtrack(vector<int>& nums, int k, int target, int cur, int start, vector<int>& used){
        if(k == 0) return true;
        if(cur == target){
            return backtrack(nums, k-1, target,
```

```
0, 0, used);
        }
        for(int i = start; i < nums.size(); i++){
            if(!used[i] && cur+nums[i] <=
target){
                used[i] = true;
                if(backtrack(nums, k, target,
cur+nums[i], i+1, used)) return true;
                used[i] = false;
            }
        }
        return false;
    }
};
```

如果有任何疑问，欢迎提出。如果有更好的解法，也欢迎告知。