一、题目说明

题目416. Partition Equal Subset Sum，给定一个只包含正整数的数组，判断数组能否分成2个子数组，且其和相等。难度是Medium！

二、我的解答

这个题目，类似背包问题，用递归算法可解。但是需要先降序排序。

```
class Solution{
    public:
        bool canPartition(vector<int>& nums){
            int sum = 0;
            int len = nums.size();

            for(int i=0;i<len;i++){
                sum += nums[i];
            }

            //如果和是奇数，不能分成2部分
            if(sum%2 !=0) return false;

            //降序排列
            sort(nums.begin(),nums.end(),greater<int>());

            sum = sum/2;
            return dfs(nums,sum,0);
        }
        bool dfs(vector<int>& nums,int sum,int index){
            if(index >= nums.size() || nums[index] > sum){
                return false;
            }
            if(nums[index] == sum){
                return true;
            }
            return dfs(nums,sum-nums[index],index+1) || dfs(nums,sum,index+1);
        }
};
```

性能如下：

```
Runtime: 0 ms, faster than 100.00% of C++ online submissions for Partition Equal Subset Sum.
Memory Usage: 8.5 MB, less than 76.47% of C++ online submissions for Partition Equal Subset Sum.
```

三、优化措施

无