

剑指57-II 和为s的连续正数序列

输入一个正整数 target，输出所有和为 target 的连续正整数序列（至少含有两个数）。序列内的数字由小到大排列，不同序列按照首个数字从小到大排列。

滑动窗口!!!

如果值小了，那么右边界右移。

如果值大了，那么左边界右移。

如果值相等，则记录，然后左边界右移。

就这么简单，因为限制了最大元素等于target的一半，所以不会存在单个元素就大于等于target的情况，所以无需判断 $i < j$ 。

像状态机一样思考！不要想复杂了去内部循环！！记住这个结构：

while() {

if() {

else if{

else {

}使用**while**灵活，不要被**for**限制住了

```
1 // 别人的优秀答案
2 vector<vector<int>> findContinuousSequence(int target) {
3     int i = 1; // 滑动窗口左边界
4     int j = 1; // 滑动窗口右边界
5     int sum = 1; // 左闭右闭区间，因此，sum的初始值为 1（也就是先不考虑right的值
6     vector<vector<int>> res;
7     // 最少两个数的情况下就是[target/2, target/2 +1]，即左边界最远最远走到target/2
8     while(i <= target / 2){
9         if(sum < target){
10             // 右边界向右移动
11             // 左闭右闭区间，先增大right,再加上right
12             j++;
13             sum += j;
14         }
15         else if(sum > target){
16             // 左边界向右移动
17             // 左闭右闭区间，先减去 left, 再left++
18             sum -= i;
19             i++;
20         }
21         else{
22             // 记录结果
23             vector<int> arr;
24             for(int k = i; k <= j; k++){
25                 arr.push_back(k);
26             }
27             res.push_back(arr);
28             // 左边界向右边移动
29             sum -= i;
30             i++;
```

```

31     }
32 }
33 return res;
34 }
35 // 如果要更优雅，记录结果的arr在外面提前定义，每次移动的时候增删arr，记录结果的时候直接push进res就行。

```

```

1 // 我的滑动窗口代码思路结构不简洁。不要使用了。
2 class Solution {
3 public:
4     vector<vector<int>> findContinuousSequence(int target) {
5         if(3 > target) return {};
6         int l = 0, r = 2;
7         int sum = 1;
8         vector<vector<int>> output;
9
10        for(; r <= target/2 + 1 ; r++) {
11            sum += r;
12            if(sum == target) {
13                vector<int> ans;
14                for(int i = l+1; i <= r; i++) {
15                    ans.push_back(i);
16                }
17                output.push_back(ans);
18            }
19            else if(sum > target) {
20                while(l <= r - 2) {
21                    l++;
22                    sum -= l;
23                    if(sum == target) {
24                        vector<int> ans;
25                        for(int i = l+1; i <= r; i++) {
26                            ans.push_back(i);
27                        }
28                        output.push_back(ans);
29                    }
30                    else if(sum < target) {
31                        break;
32                    }
33                }
34            }
35        }
36        return output;
37    }
38 };

```