1. Combination Sum III

Find all possible combinations of ***k*** numbers that add up to a number ***n***, given that only numbers from 1 to 9 can be used and each combination should be a unique set of numbers.

**Note:**

* All numbers will be positive integers.
* The solution set must not contain duplicate combinations.

**Example 1:**

Input: k = 3, n = 7  
Output: [[1,2,4]]

**Example 2:**

Input: k = 3, n = 9  
Output: [[1,2,6], [1,3,5], [2,3,4]]

**解** 定义函数f(l, r, n, target)，表示在区间[l, r]之间和为target的n个数，则：

当n=2时，直接采用双指针搜索

class Solution {  
public:  
 vector<vector<int>> combinationSum3(int k, int n) {  
 vector<bool>nums(10, true);  
 return comb(1, 9, k, n);  
 }  
 vector<vector<int>> comb(int l, int r, int n, int target){  
 vector<vector<int>>ans;  
 if(n == 2){  
 int i = l, j = r;  
 while(i < j){  
 int tmp = i + j;  
 if(tmp == target){  
 ans.push\_back({i,j});  
 i++;  
 j--;  
 }else if(tmp < target){  
 i++;  
 }else{  
 j--;  
 }  
 }  
 return ans;  
 }  
 for(int i = l; i <= r; ++i){  
 vector<vector<int>>tmp\_ans = comb(i+1, r, n - 1, target - i);  
 if(tmp\_ans.size() > 0){  
 for(auto &v : tmp\_ans){  
 v.insert(v.begin(), i);  
 ans.push\_back(v);  
 }  
 }  
 }  
 return ans;  
 }  
};