1. Permutation Sequence

The set [1,2,3,...,\*n\*] contains a total of *n*! unique permutations.

By listing and labeling all of the permutations in order, we get the following sequence for *n* = 3:

1. "123"
2. "132"
3. "213"
4. "231"
5. "312"
6. "321"

Given *n* and *k*, return the *k*th permutation sequence.

**Note:**

* Given *n* will be between 1 and 9 inclusive.
* Given *k* will be between 1 and *n*! inclusive.

**Example 1:**

Input: n = 3, k = 3  
Output: "213"

**Example 2:**

Input: n = 4, k = 9  
Output: "2314"

解法1 暴力枚举

class Solution {  
public:  
 string getPermutation(int n, int k) {  
 bool isShown[n + 1] = {false}, flag = false;  
 int cnt = 0;  
 string ans = "";  
 dfs(n, k, cnt, isShown, flag, ans);  
 return ans;  
 }  
 void dfs(int n, int k, int &cnt, bool isShown[], bool &flag, string &ans){  
 if(ans.size() == n){  
 cnt++;  
 if(cnt == k)flag = true;  
 return;  
 }  
 if(flag)return;  
 for(int i = 1; i <= n; ++i){  
 if(isShown[i] == false){  
 isShown[i] = true;  
 ans.push\_back(i + '0');  
 dfs(n, k, cnt, isShown, flag, ans);  
 if(flag)return;  
 ans.pop\_back();  
 isShown[i] = false;  
 }  
 }  
 }  
};

解法2 找规律

第1位数字，1占据 1 ~ (n - 1)! 个， 2占据 (n - 1)! + 1 ~ 2 \* (n - 1)!个，以此类推

第2位数字，1占据(n - 1)!中的1~(n-2)!个，以此类推

。。。。

class Solution {  
public:  
 string getPermutation(int n, int k) {  
 int fact[n] = {0};  
 fact[0] = 1;  
 for(int i = 1; i < n; ++i)fact[i] = i \* fact[i - 1];  
 vector<int>numbers(n);  
 for(int i = 0; i < n; ++i)numbers[i] = i + 1;  
 string ans = "";  
 k--;  
 for(int i = 1; i <= n; ++i){  
 int index = k / fact[n - i];  
 ans.push\_back(numbers[index] + '0');  
 numbers.erase(numbers.begin() + index);  
 k = k % fact[n - i];  
 }  
 return ans;  
 }  
};