

60. Permutation Sequence

题目描述: <https://leetcode.com/problems/permutation-sequence/>

给定两个数 n 和 k , 求 $1-n$ 的数组成的排列数的第 k 个
例如:

```
n = 3
k = 1  "123"
k = 2  "132"
k = 3  "213"
k = 4  "231"
k = 5  "312"
k = 6  "321"
```

解题思路:

1. 最直接的就是按照subset那种模式去求, 但是会超时
2. 数学方法:

n 个数的阶乘中, 第1位有 n 种取法, 每一种都对应 $(n-1)!$ 种

因此第 k 个的第一位应该是第 $g = k/(n-1)!$ 个数。

后面的数是第 $k = k \% (n-1)!$ 个 $(n-1)!$ 中的数

以此类推, 注意下标要对应, 因为 当 k 为下标时求出来的才是下标, 因此 ky 要先减去1

代码:

```
class Solution {
public:
    string getPermutation(int n, int k) {
        vector<char> e;
        vector<int> f(n+1, 1);
        for(int i = 1; i <= n; i++) {
            f[i] = f[i-1]*i;
        }

        string s;
        for(int i = 1; i <= n; i++) {
            e.push_back(i+'0');
        }
        k = k - 1;
        for(int i = 0; i < n; i++) {
            int g = k / f[n-i-1];
            k = k % f[n-i-1];
            s.push_back(e[g]);
            e.erase(e.begin()+g);
        }
        return s;
    }
};
```