Permutations

Given a collection of distinct integers, return all possible permutations.

对问题进行观察，考虑是否可以reduce原问题？考虑input的数据类型为数组：

- 直接将数组一分为二，对问题求解没有帮助
- 从小规模的问题出发，当n=1,n=2,n=3时，从中可以得出一个规律：对于k个已得出解的数字，再加入一个与前k个数相异的数a_k,将解中第1至第k个元素与a_k依次交换位置，则可以得到k+1个数的解

由该观察可以发现该问题满足最优子结构的性质，可以用动态规划求解，用$OPT(k)$表示前k个数的解(代表前k个数符合题意的所有排列)，可以得到下述递归表达式；

$$OPT(k+1) = swap(a_{k+1}, enumerate\ \ all\ \ elements\ \ in\ \ OPT(k))$$

*对于求解思路，请参考算法笔记（一）概述中的**思路树***

GitHub地址

```
class Solution {

private:
    void getOrder(vector<vector<int>>& result,vector<int> currentNums,int low,int high){
        if(low==high){
            result.push_back(currentNums);
            return;
        }
        for(int i=0;i<=low;++i){
            swap(currentNums[i],currentNums[low]);
            getOrder(result,currentNums,low+1,high);
            swap(currentNums[i],currentNums[low]);
        }
    }
public:

    vector<vector<int>> permute(vector<int>& nums) {
        vector<vector<int>> result;
        getOrder(result,nums,1,nums.size());
        return result;
    }

```};
```
```