

# 046. Permutations

## 046 Permutations

- BackTracking+array

### Description

Given a collection of **distinct** numbers, return all possible permutations.

For example,

`[1,2,3]` have the following permutations:

```
[
  [1,2,3],
  [1,3,2],
  [2,1,3],
  [2,3,1],
  [3,1,2],
  [3,2,1]
]
```

### 1. Thought line

### 2. BackTracking+array

```
1 class Solution {
2 private:
3     void backTracking_fct(vector<vector<int>>& result, vector<int>& nums, vector<bool>& flag,
4                           vector<int>& temp){
5         if (temp.size()==nums.size()){
6             result.push_back(temp);
7             return;
8         }
9
10        for (int i = 0; i<=flag.size()-1; ++i){
11            if (!flag[i]){
12                temp.push_back(nums[i]);
13                flag[i] = true;
14                backTracking_fct(result, nums, flag, temp);
15                flag[i] = false;
16                temp.pop_back();
17            }
18        }
19    }
20
21 public:
22     vector<vector<int>> permute(vector<int>& nums) {
23         vector<vector<int>> result;
24         vector<bool> flag(nums.size(),false);
25         vector<int> temp;
26         backTracking_fct(result, nums, flag, temp);
27         return result;
28     }
29 };
30 };
```