**46. Permutations :-**

Medium Accepted: 1.7M Submissions: 2.2M Acceptance Rate: 76.5%

Given an array nums of distinct integers, return *all the possible permutations*. You can return the answer in **any order**.

**Example 1:**

**Input:** nums = [1,2,3]

**Output:** [[1,2,3],[1,3,2],[2,1,3],[2,3,1],[3,1,2],[3,2,1]]

**Example 2:**

**Input:** nums = [0,1]

**Output:** [[0,1],[1,0]]

**Example 3:**

**Input:** nums = [1]

**Output:** [[1]]

**Constraints:**

* 1 <= nums.length <= 6
* -10 <= nums[i] <= 10
* All the integers of nums are **unique**.

**Code :-**

class Solution {

public:

    void func(vector<int> &num, int ind, vector<vector<int>> &ans, vector<int> &temp){

        int n=num.size();

        if(ind>n){

            ans.push\_back(temp);

            return;

        }

        for(auto i=0; i<n; i++){

            if(num[i] != INT\_MAX){

                temp.push\_back(num[i]);

                num[i] = INT\_MAX;

                func(num, ind+1, ans, temp);

                num[i] = temp.back();

                temp.pop\_back();

            }

        }

        return;

    }

    vector<vector<int>> permute(vector<int>& num) {

        vector<vector<int>> ans;

        vector<int> temp;

        func(num, 1, ans, temp);

        return ans;

    }

};