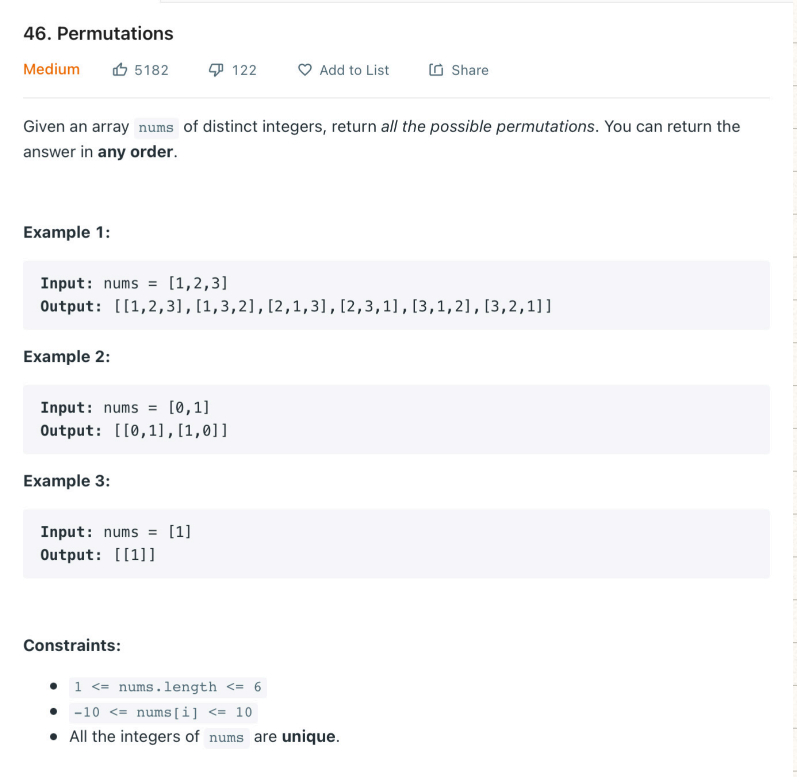


46. Permutations

Medium 5182 122 Add to List Share

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:

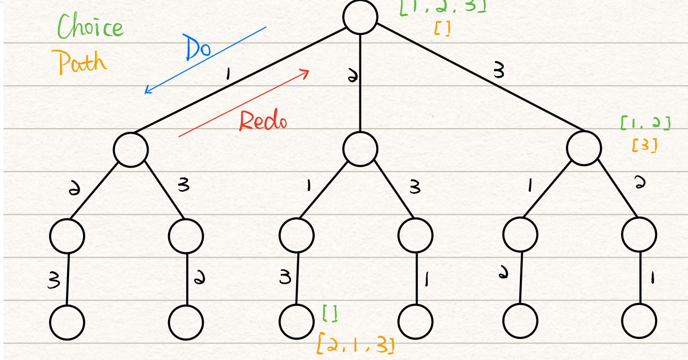
```
Input: nums = [0,1]
Output: [[0,1],[1,0]]
```

Example 3:

```
Input: nums = [1]
Output: [[1]]
```

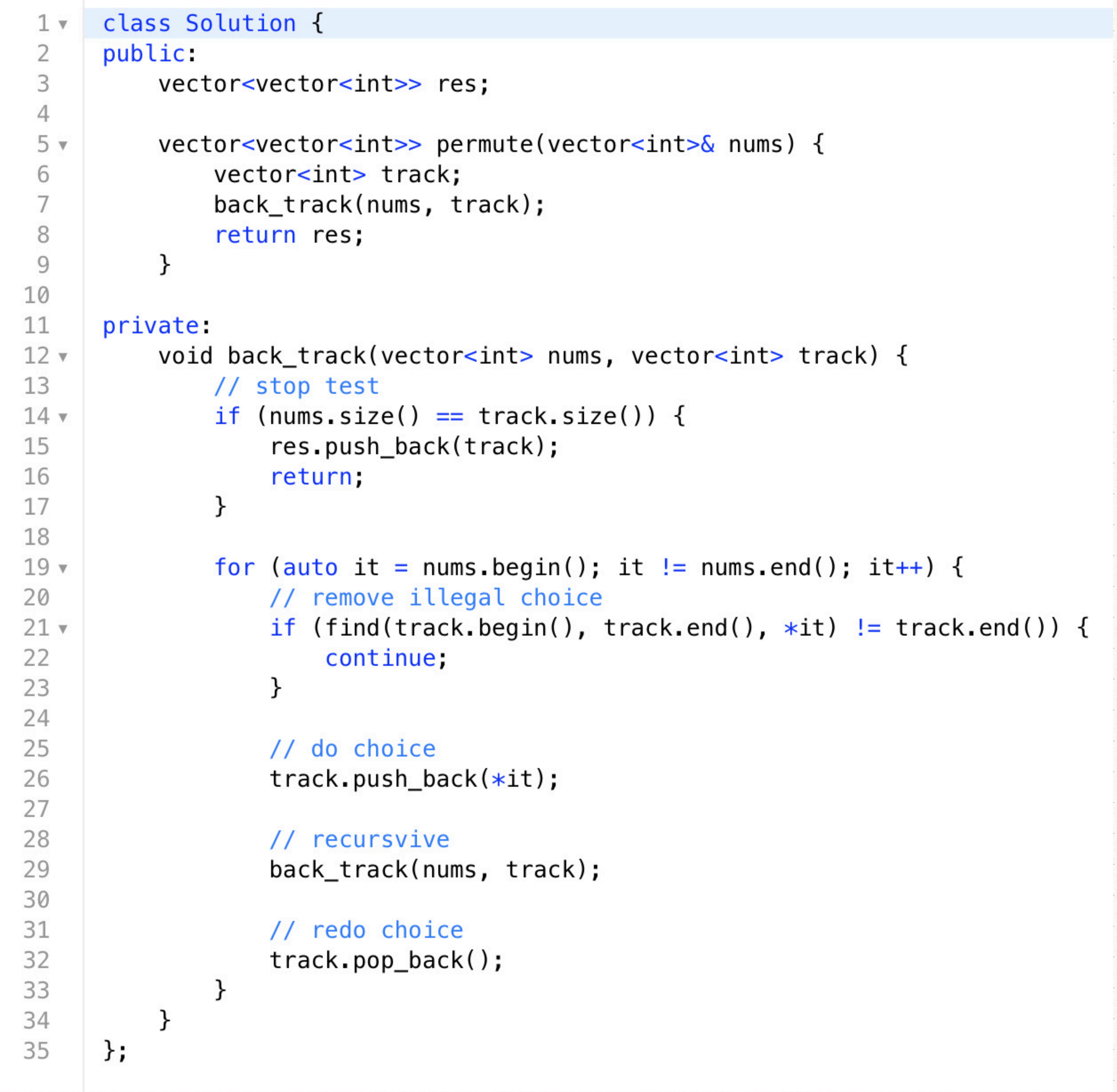
Constraints:

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



Time Complexity : $O(n^2)$

Space Complexity : $O(n)$



```cpp
class Solution {
public:
    vector<vector<int>> res;

    vector<vector<int>> permute(vector<int>& nums) {
        vector<int> track;
        back_track(nums, track);
        return res;
    }

private:
    void back_track(vector<int> nums, vector<int> track) {
        // stop test
        if (nums.size() == track.size()) {
            res.push_back(track);
            return;
        }

        for (auto it = nums.begin(); it != nums.end(); it++) {
            // remove illegal choice
            if (find(track.begin(), track.end(), *it) != track.end()) {
                continue;
            }

            // do choice
            track.push_back(*it);

            // recursvive
            back_track(nums, track);

            // redo choice
            track.pop_back();
        }
    }
};
```