46. Permutations

Medium

163146FavoriteShare

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

Example:

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

Accepted

332,033

Submissions

626,767

CPP: BackTracking

class Solution {

public:

vector<int> input\_vector;

vector<vector<int>> return\_vector;

bool contains(vector<int> vec,int x){

if(std::count(vec.begin(),vec.end(),x)>0) return true;

return false;

}

void BackTrack(vector<int> param){

if(param.size()==input\_vector.size()){

return\_vector.push\_back(param);

return;

}

for(int i=0;i<input\_vector.size();i++){

vector<int> param\_copy=param;

if(!contains(param\_copy,input\_vector[i])){

param\_copy.push\_back(input\_vector[i]);

BackTrack(param\_copy);

}

}

}

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

input\_vector=nums;

vector<int> bt;

BackTrack(bt);

return return\_vector;

}

};

Success

[Details](https://leetcode.com/submissions/detail/207499242/)

Runtime: 32 ms, faster than 2.93% of C++ online submissions forPermutations.

Memory Usage: 12.6 MB, less than 0.81% of C++ online submissions forPermutations.

CPP: std::next\_permutation

class Solution {

public:

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

vector<vector<int>> ret;

std::sort(nums.begin(),nums.end());

do{

ret.push\_back(nums);

}while(std::next\_permutation(nums.begin(),nums.end()));

return ret;

}

};

Success

[Details](https://leetcode.com/submissions/detail/207503777/)

Runtime: 16 ms, faster than 100.00% of C++ online submissions forPermutations.

Memory Usage: 9.1 MB, less than 0.81% of C++ online submissions forPermutations.