

Date : 13th May

Mentor : Gladden Rumao

Topic : Backtracking

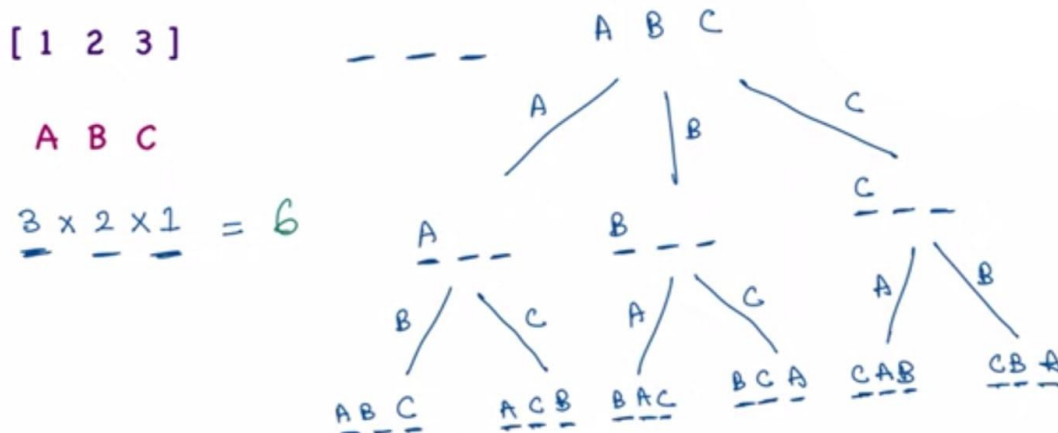
Q1. Given an array nums of distinct integers, return all the possible permutations. You can return the answer in any order.

Problem Link : <https://leetcode.com/problems/permutations/>

Approach :

- We use backtracking to generate all permutations
backtrack function generates all permutations starting from index first
- In each recursive call, we place the i-th integer first in the permutation and use the next integers to complete the permutation
- Finally, we backtrack to get the next permutation
- We keep track of all permutations in the output list and return it in the end

UNDERSTANDING THE LOGIC



Solution :

```
class Solution {
    public void backtrack(List<List<Integer>> result ,
List<Integer> temp ,int[] nums){
        if(temp.size()==nums.length){
            result.add(new ArrayList<>(temp));
            return;
        }
        for(int i=0 ; i<nums.length ; i++){
            if(temp.contains(nums[i]))
                continue;
            temp.add(nums[i]);
            backtrack(result,temp,nums);
            temp.remove(temp.size()-1);
        }
    }
    public List<List<Integer>> permute(int[] nums) {
        List<List<Integer>> result = new ArrayList<>();
        backtrack(result,new ArrayList<Integer>(),nums);
        return result;
    }
}
```