46.Permutations

46. Permutations

Medium

7225143Add to ListShare

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**.

```
class Solution:
    def permute(self, nums: List[int]) -> List[List[int]]:
        res = []
        visited = [0]*len(nums)
        self.permuteUtil(nums,res,[],0,visited)
        return res
    def permuteUtil(self,nums,res,asf,te,visited):
        if te==len(nums):
            temp = asf[:]
            res.append(temp)
            return
        for i in range(len(nums)):
            if visited[i]==0:
                visited[i] = 1
                self.permuteUtil(nums,res,asf+[nums[i]],te+1,visited)
                visited[i]=0
```