class Solution(object):

def permute(self, nums):

visited = set()

res = []

self.backtracking(res,visited,[],nums)

return res

"""

Level0: []

level1: [1] [2] [3]

level2: [1,2] [1,3] [2,1] [2,3] [3,1] [3,2]

level3: [1,2,3] [1,3,2] [2,1,3][2,3,1] [3,1,2][3,2,1]

"""

def backtracking(self,res,visited,subset,nums):

if len(subset) == len(nums):

res.append(subset)

for i in range(len(nums)):

if i not in visited:

visited.add(i)

self.backtracking(res,visited,subset+[nums[i]],nums)

visited.remove(i)