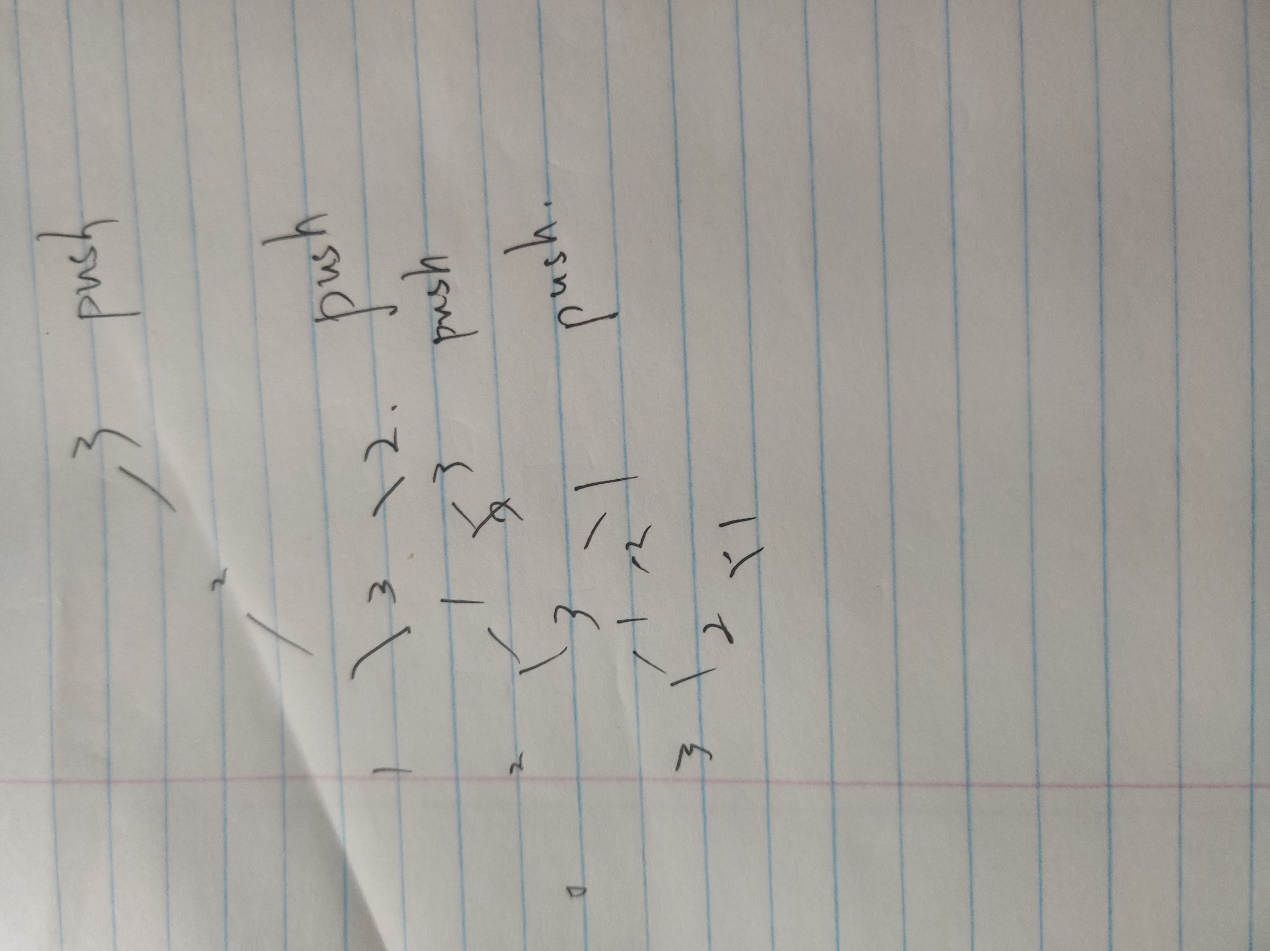


典型的DFS题目

不过这次不用index，因为我们要找出所有而不是顺序，2有可能在1的前面，但总体思路还是一样的



可以看见提取过以后就不会再用那个元素了，因此我们可以用一个arrayList来循环，先把这个元素Push到curr,再从候选项中移除这个元素

class Solution {

public static List<List<Integer>> permute(int[] nums) {

List<List<Integer>> result=new ArrayList<>();

List<Integer> list = new ArrayList<>();

List<Integer> curr = new ArrayList<>();

for (int i:nums){

list.add(i);

}

permuteRecursive(list,curr,result);

return result;

}

public static void permuteRecursive(List<Integer> nums,List<Integer> curr,List<List<Integer>> result){

if(nums.size()==0){

result.add(new ArrayList<>(curr)); //candidate=0代表排列完了

return;

}

for(int i=0;i<nums.size();i++){

int temp=nums.get(i); //提取这个元素加入curr

curr.add(temp);

nums.remove(i); //暂时移除

permuteRecursive(nums,curr,result);

curr.remove(curr.size()-1); currPop出来给下一轮循环用

nums.add(i,temp); //nums移回这个数因为为我们下一轮还能取

}

}

}