**Step 1:-Understand the problem**

1. Can we restate the problem in our own words?

**Find the number of ways in which the given array can be ordered**

1. What are the inputs that go into the problem?

**Arrays of type int**

1. What are the outputs that come from the problem?

**Lists**

1. Can the outputs be determined from the inputs? In other words do we have enough information to solve this problem?

**yes**

1. What should I label the important piece of data that are the part of a problem?

**Already labelled**

**Step 2:-Explore examples**

1. Start with simple examples

[1,2]

**[1,2],[2,1]**

**2)**

**[1,2,3]**

**[1,2,3],[1,3,2],[2,1,3],[2,3,1],[3,1,2],[3,2,1]**

1. Progress to more complex examples

**[1,2,3,4]**

**[1,2,3,4]**

**[2,1,3,4]**

**[2,3,1,4]**

**[2,3,4,1]**

**[3,2,4,1]**

**[3,4,2,1]**

**[3,4,1,2]**

**[4,3,1,2]**

**[4,1,3,2]**

**[4,1,2,3]**

**[1,4,2,3]**

**[1,2,4,3]**

1. Explore examples with empty
2. Explore the examples with invalid inputs

**Step 3:-Break it down**

**Step 4:-Solve/Simplify**

* Find the core difficulty
* Temporarily ignore that difficulty
* Write a simplified solution
* Then incorporate that difficulty

**Step 5:-Look back refractor**

1. Can we check the result?
2. Can we drive the result differently?
3. Can we understand it at a glance?
4. Can we use the result or method for some other problem?
5. Can you improve the performance of your solution?
6. How other people solve this problem?

import math

class Solution:

def permute(self, nums: List[int]) -> List[List[int]]:

import math

temp\_vari=0

l=[]

if len(nums)==1:

return [nums]

for \_ in range(int(math.factorial(len(nums))/(len(nums)-1))):

for i in range(0,len(nums)-1):

temp\_vari=nums[i]

nums[i]=nums[i+1]

nums[i+1]=temp\_vari

a=nums.copy()

l.append(a)

return l

The above code is not working for 4 numbers

Swap the corresponding digits n times

123

213

231

And so on…………