46. Permutations

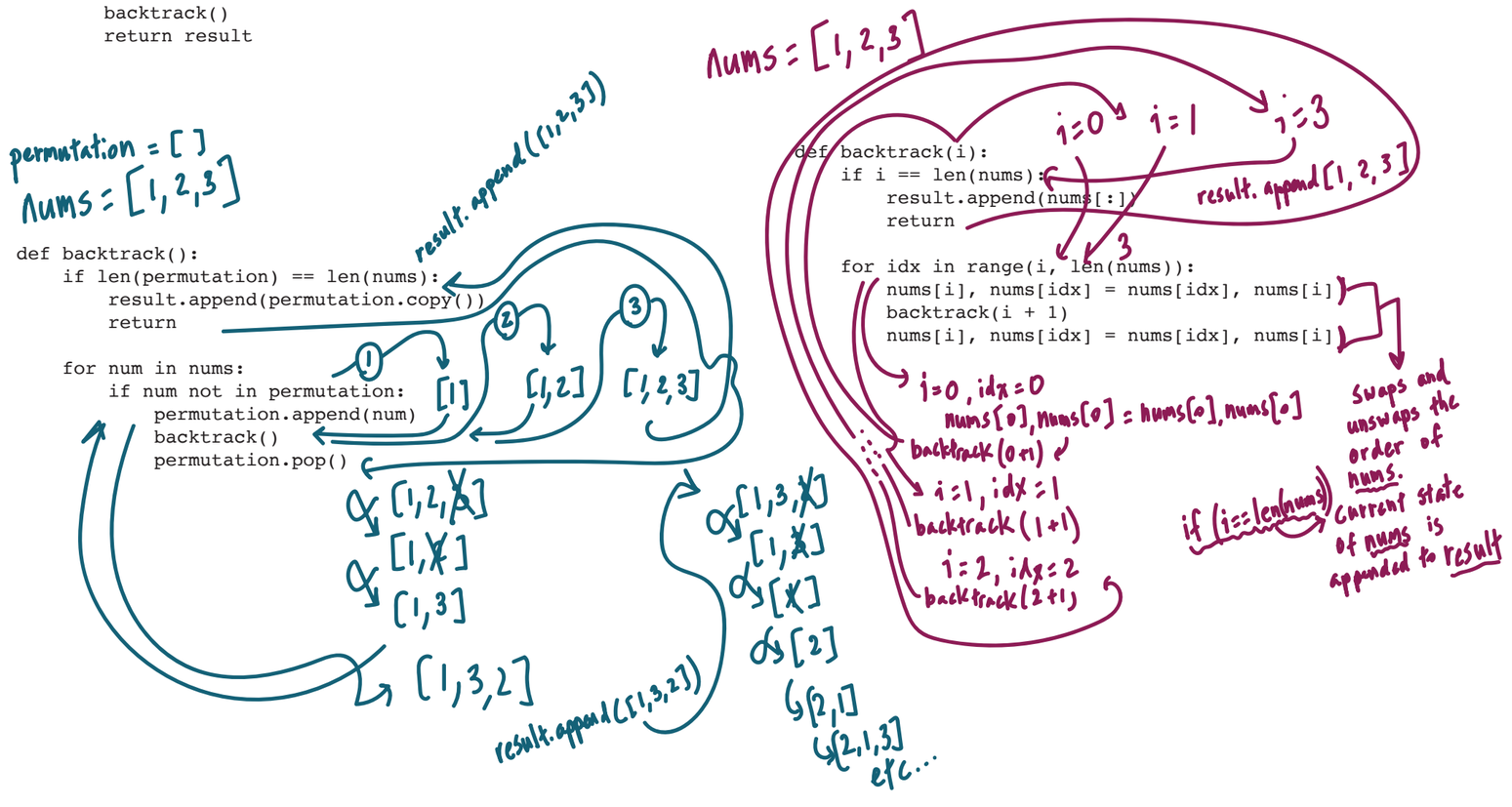
Medium

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

```
class Solution:
    def permute(self, nums: List[any]) -> List[List[any]]:
        result = []
        permutation = []

        def backtrack():
            if len(permutation) == len(nums):
                result.append(permutation.copy())
                return

            for num in nums:
                if num not in permutation:
                    permutation.append(num)
                    backtrack()
                    permutation.pop()

        backtrack()
        return result
```

```
class Solution:
    def permute(self, nums: List[int]) -> List[List[int]]:
        result = []

        def backtrack(i):
            if i == len(nums):
                result.append(nums[:])
                return

            for idx in range(i, len(nums)):
                nums[i], nums[idx] = nums[idx], nums[i]
                backtrack(i + 1)
                nums[i], nums[idx] = nums[idx], nums[i]

        backtrack(0)
        return result
```

```
class Solution:
    def permute(self, nums: List[any]) -> List[List[any]]:
        result = []

        def backtrack(nums_list, current_permutation):
            if not nums_list:
                result.append(current_permutation[:])
                return

            for i in range(len(nums_list)):
                n = nums_list[i]
                current_permutation.append(n)
                backtrack(nums_list[:i] + nums_list[i+1:], current_permutation)
                current_permutation.pop()

        backtrack(nums, [])
        return result
```




```
def backtrack(nums_list, current_permutation):
    if not nums_list:
        result.append(current_permutation[:])
        return

    for i in range(len(nums_list)):
        n = nums_list[i]
        current_permutation.append(n)
        backtrack(nums_list[:i] + nums_list[i+1:], current_permutation)
        current_permutation.pop()
```

if nums_list is empty