①

```
def permutations(n):
    permutationsList = []
    if n == 1: return [[1]]        # base case

    for i in range(n):
        for permutation in permutations(n-1):
            permutation.insert(i, n)
            permutationsList.append(permutation)

    return permutationsList
```

the magic

the magic: every permutation is a rearrangement of n (from first to last element in a list)

for every rearrangement of n-1

②

```
def push(self, element):
    self.queue.enqueue(element)
    while self.queue.top() != element:
        self.queue.enqueue(self.queue.pop())
def pop(self):
    self.queue.pop()
def top(self):
    self.queue.top()
def size(self):
    i = -1
    top = top(self)
    self.queue.enqueue(self.queue.pop())
    while True:
        self.queue.enqueue(self.queue.pop())
        i += 1
        if top == top(self): break;
    return i
def is_empty(self):
    return size(self) == 0
```