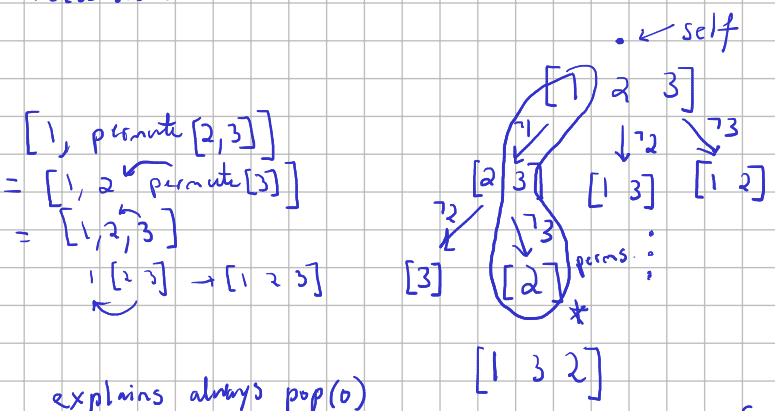
Leetcode 46: Permutations

Backtracking, Recursion

[1, 2, 3] → all 3!



permute(→ def permute

base case

if len(nums) ≤ 1

return [nums]

)

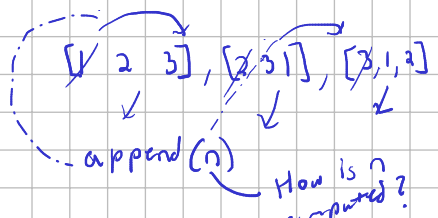
[1, permute [2,3]]

= [1, 2 permute [3]]

= [1, 2, 3]

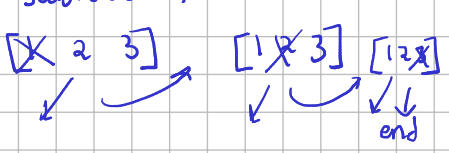
[2 3] → [1 2 3]

explains always pop(0)



=

Subroutine:



for i in range(len(nums))

end

dfs style recursion in call stack

self.perms = permute(nums)

n = nums.pop(0)

ret value of recursive call

getting perm

for perm in perms:

perm.append(n)

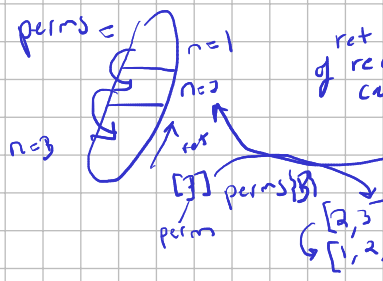
end

ret perm to res[]

result.extend(perms)

nums.append(n)

end



Extend vs. Append function?

[2 3].extend(1) = [2 3 1]

[2 3].append(1) = [1 2 3]