

1 1 0 0 0

1 0 0 1

1 0 1

1 0

1









**###### Phone Number Letter Combinations #####**

****

if (nums[e] in dictt) and (dictt[nums[e]] > 0):

dictt[nums[e]] -= 1

end += 1

else:

while start < end:

dictt[nums[start]] += 1

## Permutations (no duplicates in list)



self.num2char = [[0], [1], ['a', 'b', 'c'], …]

def dfs(self, l\_count, r\_count, path):

if len(path) == self.n:

out = ''.join(path)

return out

else:

for c in self.num2char[self.nums[index]]:

self.dfs(index + 1, path + ['c'])

def solve(self, nums):

self.n = len(nums)

self.out = []

# nums = sorted(nums)

self.nums = nums

self.dfs(index, [])

return self.out

