1. Sort Colours

class Solution(object):

def sortColors(self, nums):

n = len(nums)

for i in range(n):

swap = False

for j in range(0, n - i - 1):

if nums[j] > nums[j + 1]:

nums[j], nums[j + 1] = nums[j + 1], nums[j]

swap = True

if swap == False:

break

return nums

1. Subsets

class Solution(object):

def subsets(self, nums):

lists = [[]]

for i in range(len(nums) + 1):

for j in range(i):

lists.append(nums[j: i])

ln = len(lists)

for i in range(ln):

swap = False

for j in range(0, ln - i - 1):

if len(lists[j]) > len(lists[j + 1]):

lists[j], lists[j + 1] = lists[j + 1], lists[j]

swap = True

if swap == False:

break

return lists

1. Merge Sorted Array

class Solution(object):

def merge(self, nums1, m, nums2, n):

lists = []

for i in nums1:

if i == 0:

continue

else:

lists.append(i)

for j in nums2:

if j == 0:

continue

else:

lists.append(j)

ln = len(lists)

for i in range(ln):

swap = False

if lists[i] == 0:

continue

else:

for j in range(0, ln - i - 1):

if lists[j] > lists[j + 1]:

lists[j], lists[j + 1] = lists[j + 1], lists[j]

swap = True

if swap == False:

break

return lists

1. Longest Consecutive Sequence

class Solution(object):

def longestConsecutive(self, nums):

ans = []

n = len(nums)

if n > 0:

for i in range(n):

swap = False

for j in range(0,n-i-1):

if nums[j]>nums[j+1]:

nums[j],nums[j+1] = nums[j+1],nums[j]

swap = True

if swap == False:

break

ans.append(nums[0])

for k in range(1,n):

if nums[k]==(nums[k-1]+1) and nums[k] not in ans and nums[k]>=0:

if nums[k]==(ans[-1]+1):

ans.append(nums[k])

else:

ans \*= 0

ans.append(nums[k-1])

ans.append(nums[k])

return len(ans)

return 0