Python Examples:

1. Given an array of length n, all the numbers in the array are between 0 and n-1. Assume there is duplicate numbers in the array. Find out all the missing numbers in the array. Requirement: we are not allowed to use extra space. Please work on the array itself.

E.g, arr = [2,1,2,2,3], length = 5, missing = [0,4]

def notshown(arr):

res = []

for i in range(0, len(arr)):

idx = abs(arr[i])

if arr[idx] > 0:

arr[idx] = -1\*arr[idx] # marked as visited

for i in range(0, len(arr)):

if arr[i] > 0:

res.append(i)

return res

2. Combination sum.

Given a set of candidate numbers (candidates) (without duplicates) and a target number (target), find all unique combinations in candidates where the candidate numbers sums to target.

The same repeated number may be chosen from candidates unlimited number of times.

Note:

* All numbers (including target) will be positive integers.
* The solution set must not contain duplicate combinations.

Example 1:

Input: candidates = [2,3,6,7], target = 7,

A solution set is:

[

[7],

[2,2,3]

]

Example 2:

Input: candidates = [2,3,5], target = 8,

A solution set is:

[

[2,2,2,2],

[2,3,3],

[3,5]

]

def combinationSum(arr, tgt):

result = []

if tgt < 0:

return result

for i in range(0, len(arr)):

if tgt - arr[i] > 0:

tmp\_result = combinationSum(arr, tgt-arr[i])

if len(tmp\_result) > 0:

tmp\_result = [ele + [arr[i]] for ele in tmp\_result]

result.extend(tmp\_result)

elif tgt - arr[i] == 0:

result.append([arr[i]])

return result

combinationSum(candidates, 7)

3. Flatten list. A list may contain another list as its element. Flatten the list.

def flatten\_list(lst):

res = []

def flatten(sublst):

for ele in sublst:

if isinstance(ele, list):

flatten(ele)

else:

res.append(ele)

flatten(lst)

return res

a = [[[1,2]], [3,4]]

print(flatten\_list(a))