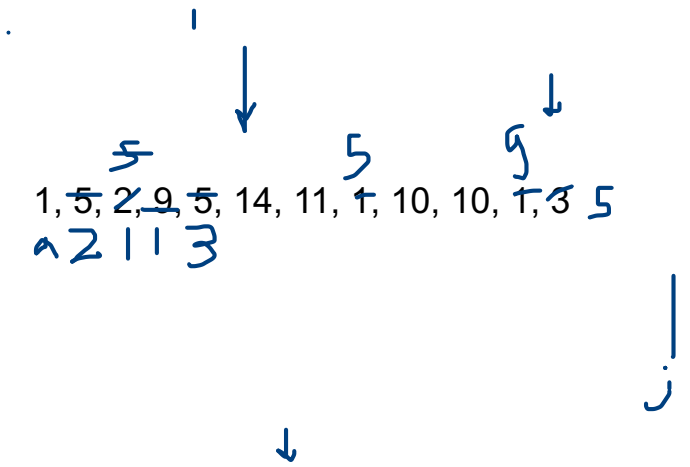
Segregate elements on pivot



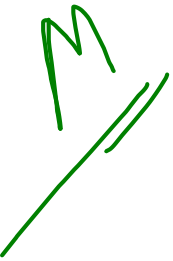
1, 2, 1, 1, 3, 14, 11, 5, 10, 10, 9, 5

if (j.val > pivot)

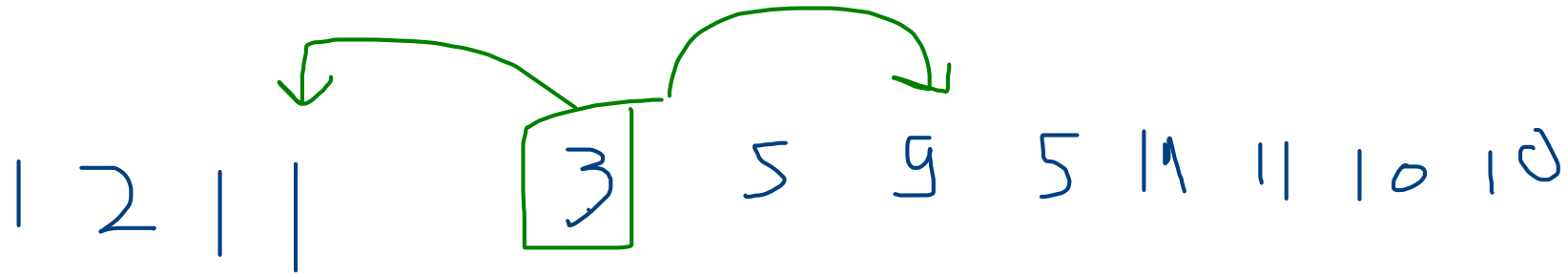
j = j.next

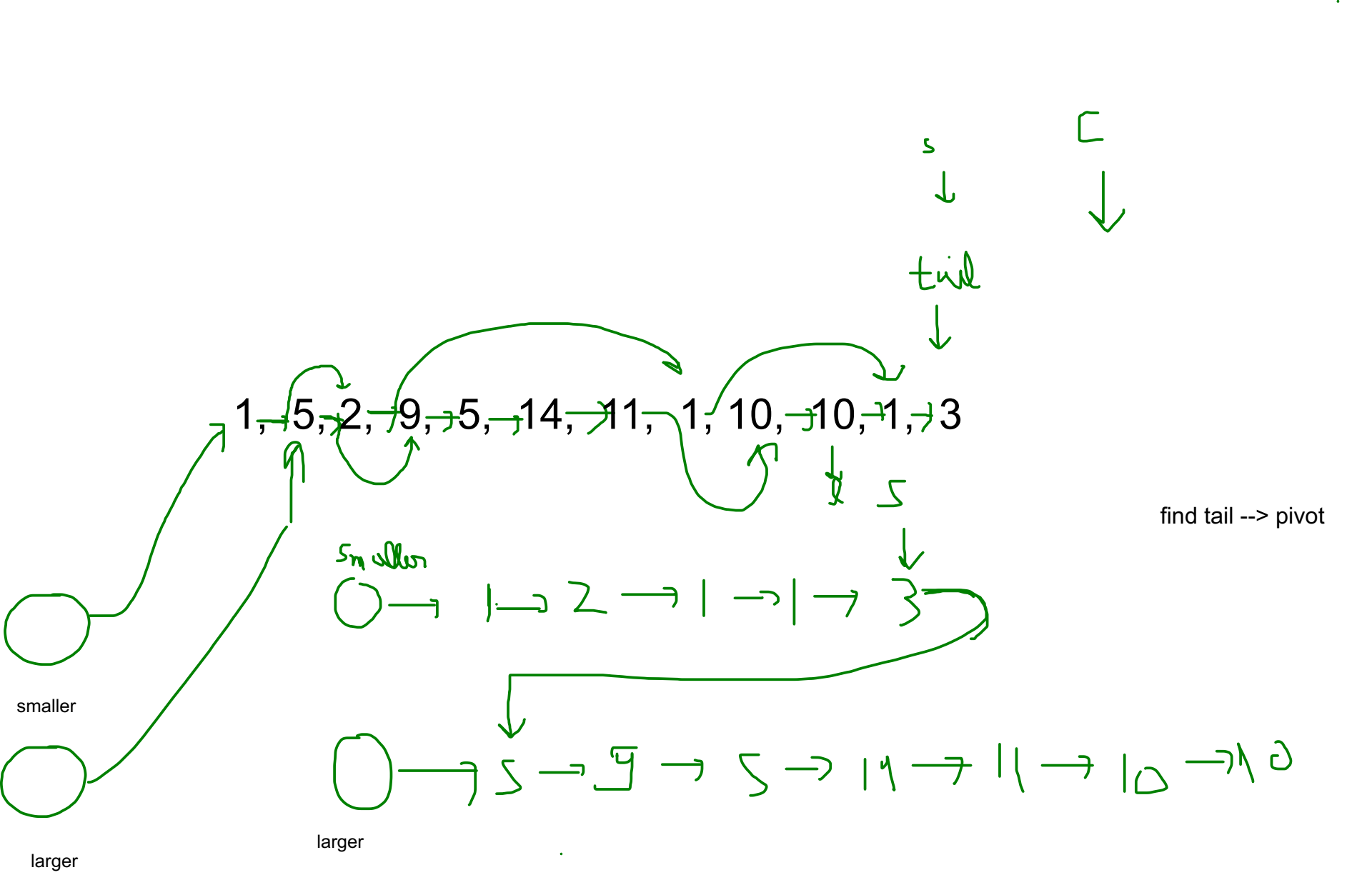
Order of element is changed

if (j.val <- pivot)
i = i.next
j = next;



1, 5, 2, 9, 5, 14, 11, 1, 10, 10, 1, 3

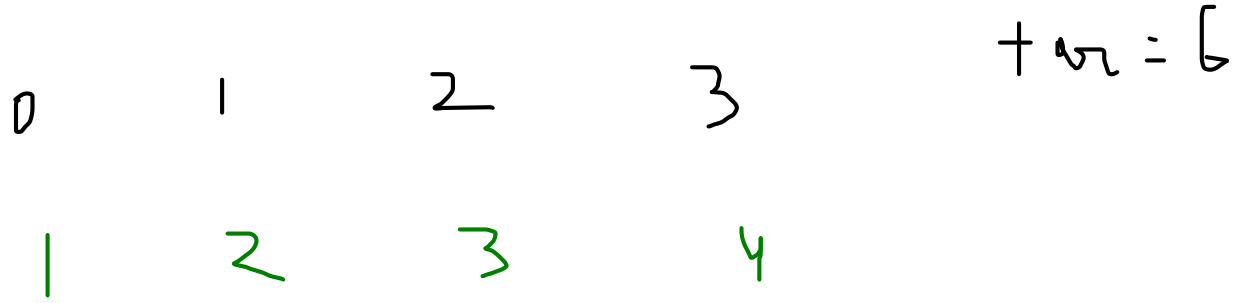




find tail --> pivot

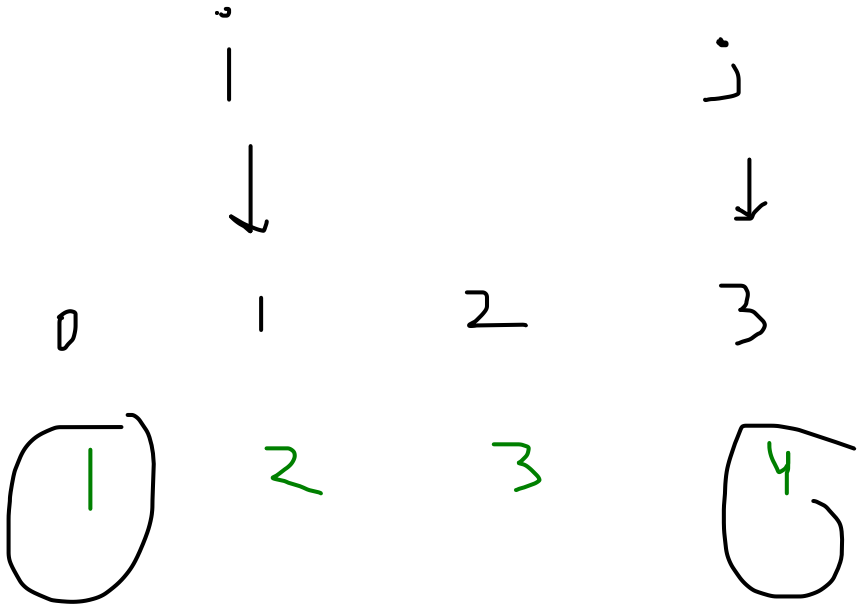
Two Pointers

Two Sum with sorted array



sorted
only one solution possible

return contant size 2 array [idx1, idx2]



i = 0
j = n - 1

tar = 6

tar > sum(i,j) --> left pointer move

tar < sum(i, j) --> right pointer

tar == sum --> stor ans

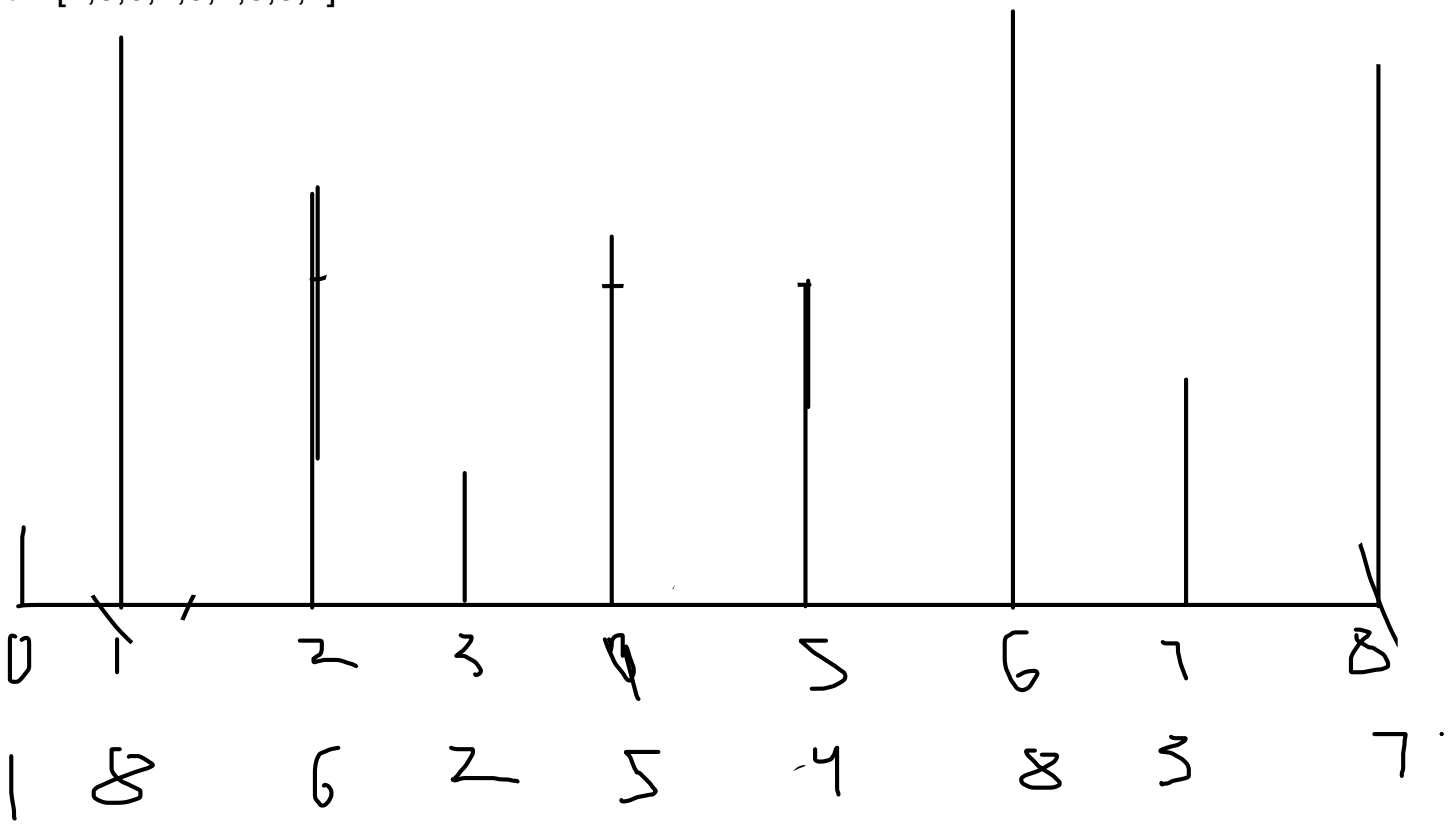
arr[i] + arr[j] = sum

1 + 4 = 5 < 6

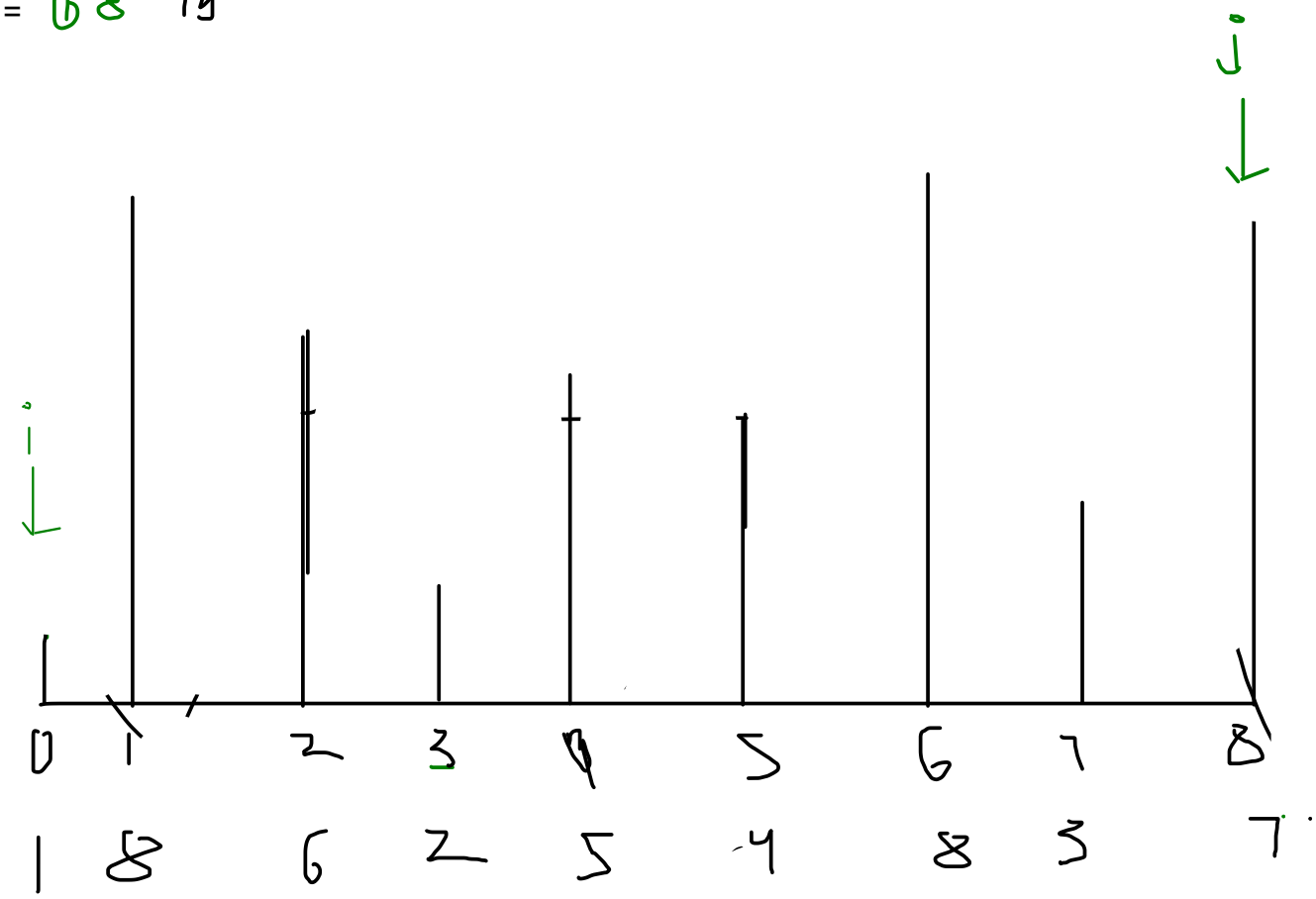
1 + 4 = 6

Container with most water

height = [1,8,6,2,5,4,8,3,7]



waterStores = 0 8 49



if (arr[i] <= arr[j])
update ans
i++

if (arr[i] > arr[j]
update ans
j--

3 Sum

0	1	2	3	4	5
-1	0	1	2	-1	-4

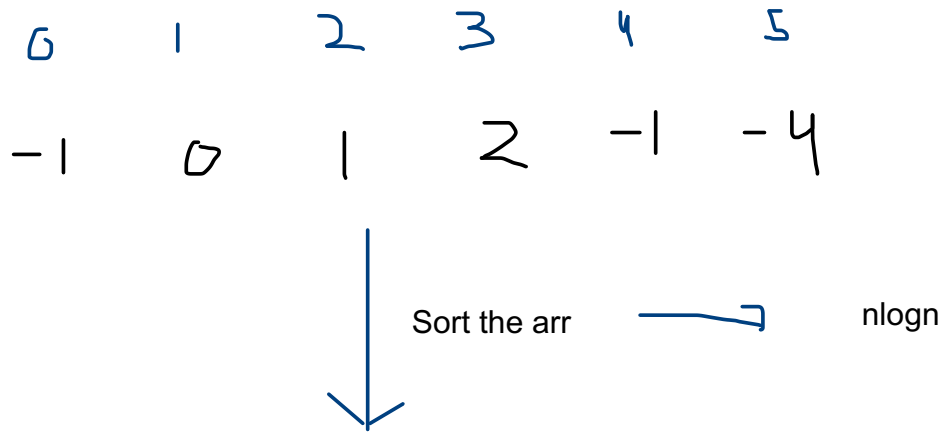
i = [0 to n-2]

j = [i to n-1]

k = [i to n]

n^3

arr[i] + arr[j] = arr[k] == 0 -> record ans



arr[i] + arr[j] + arr[k] = 0

arr[j] + arr[k] = - arr[i]

tar

two sum

n^2 + nlogn = n^2 (in BigO)