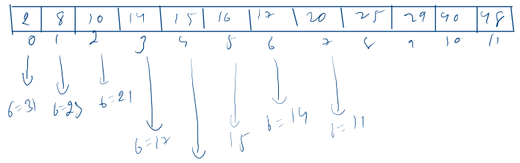
Given a **1-indexed** array of integers `numbers` that is already ***sorted in non-decreasing order***, find two numbers such that they add up to a specific `target` number. Let these two numbers be `numbers[index1]` and `numbers[index2]` where `1 <= index1 < index2 <= numbers.length`.

Return *the indices of the two numbers,* `index1` *and* `index2`*, **added by one** as an integer array* `[index1, index2]` *of length 2.*

The tests are generated such that there is **exactly one solution**. You **may not** use the same element twice.

Your solution must use only constant extra space.



S: O(1)

target: 91

arr: [3n, 6n]

a + b = target

target = 6

```python
def twoSum(numbers: List[int], target: int):
    l = len(numbers)
    for i in range(l):
        b = target - numbers[i]
        for j in range(i + 1, l):
            if b == numbers[j]:
                return [i + 1, j + 1]

    return []
```

1 → n-1
2 → n-2
3 → n-3
4 → n-4
5 → n-5

(n-1)+(n-2)+(n-3)+(n-4)+(n-5)+ ... +2+1

≤ i

T: $\frac{(n)(n-1)}{2}$

$\sim \frac{n^2}{2} - \frac{n}{2}$

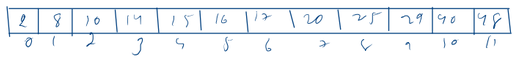
$\sim O(n^2)$

6 9

(2, 29)

(4, 13)

(5, 16)

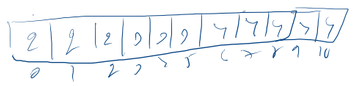


dictionary: 2 − 0
8 − 1
10 − 2
14 − 3
15 − 4

2 + 48 = 50
2 + 40 = 42

O(11) ≃ O(n)

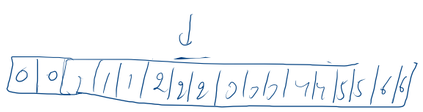
Θ(11)



target = 6

[0,10] [3,5]

target = 8
target = 6



(0,6) (1,5) (2,3) (3,3)

```python
def twoSum(arr, tar, si, ei):
    ans = []
    while si < ei:
        sum = arr[si] + arr[ei]

        if sum == tar:
            ans.append([si, ei])
            si += 1
            ei -= 1

            while si < ei and arr[si] == arr[si - 1]:
                si += 1
            while si < ei and arr[ei] == arr[ei + 1]:
                ei -= 1

        elif sum < tar:
            si += 1
        else:
            ei -= 1

    return ans
```

2 - Fix

[(2,4), (3,3)]

T: O(n)

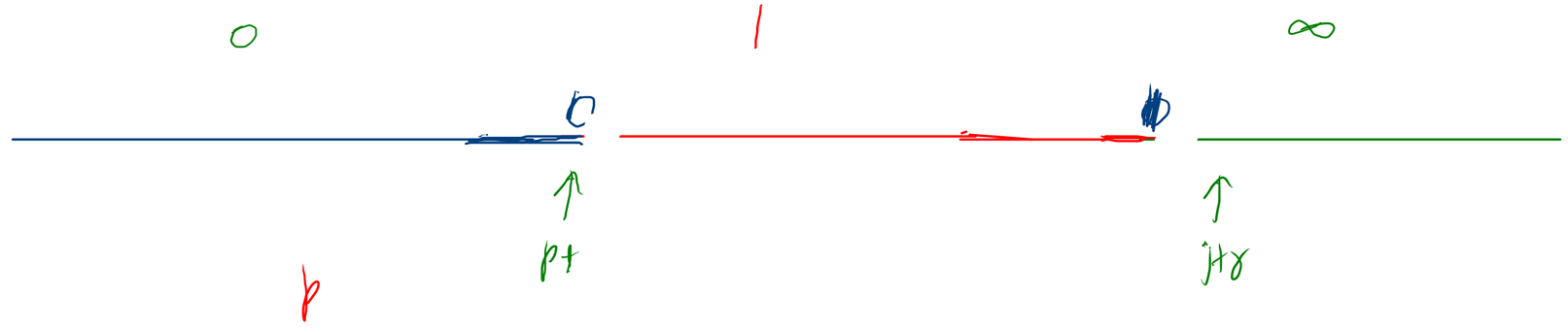
S: O(1)

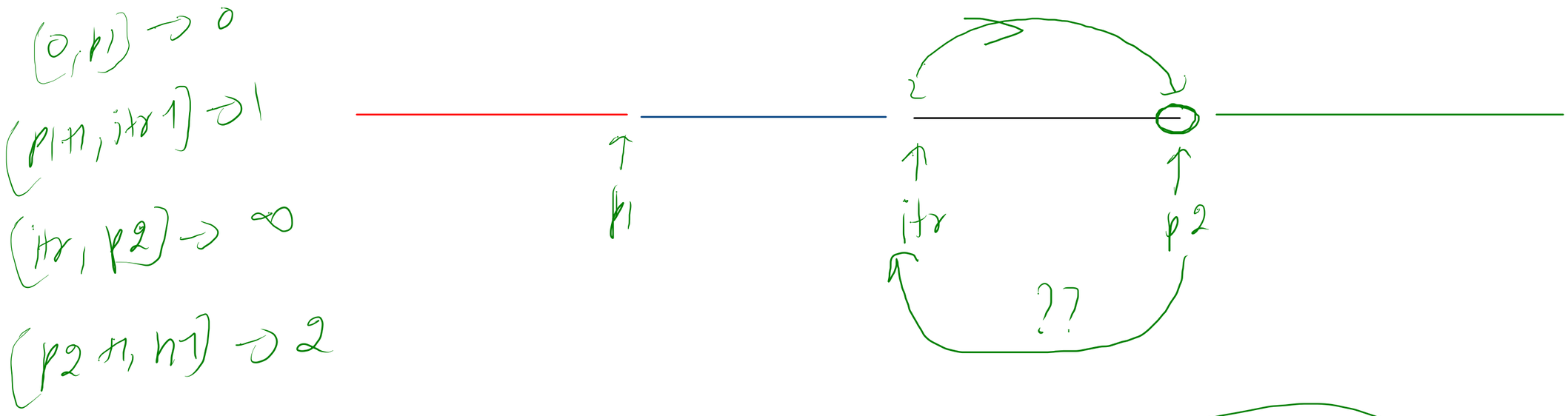
sort (0,1)

[0, pt] → 0

[pt+1, itr-1] → 1

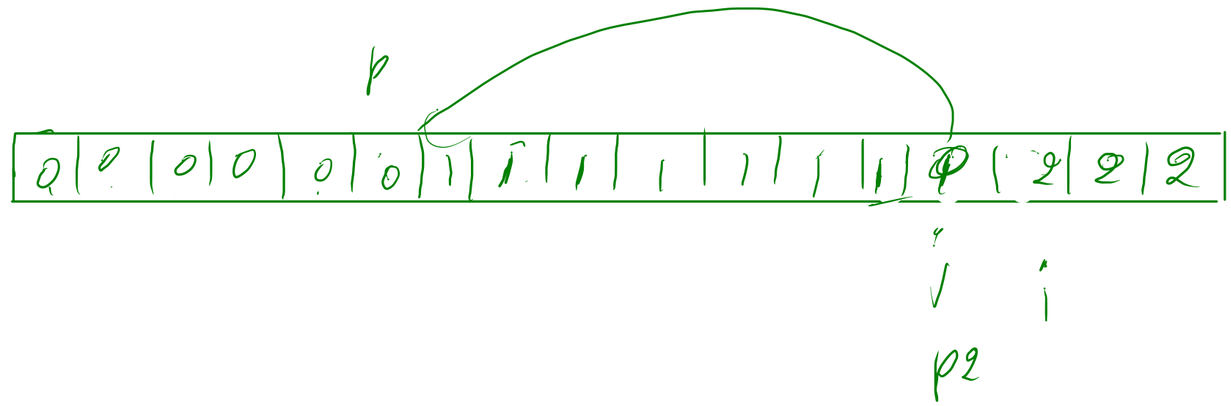
[itr, n-1] → ∞






```python
def sortColors(self, arr: List[int]) -> None:
    itr = 0
    pt = -1
    n = len(arr)
    pt2 = n - 1

    while itr < n:
        if arr[itr] == 0:
            pt += 1
            arr[pt], arr[itr] = arr[itr], arr[pt]
            itr += 1
        elif arr[itr] == 1:
            itr += 1
        else:
            arr[pt2], arr[itr] = arr[itr], arr[pt2]
            pt2 -= 1
```



(T: 7:30 - 10:30
S: 10-1