Santhosh 04-Nov

04 November 2024 18:40

→ finding all sub arrays of an array

[1, 2, 3, 4]

all subarrays:

- (1), (1, 2), (1, 2, 3), (1, 2, 3, 4)
- (2), (2, 3), (2, 3, 4)
- (3), (3, 4)
- (4)

→ 1
1, 2
1, 3

2,
2, 3
2, 3, 4

3,
3, 4

---

find all subarrays
→ compute the sum
(= k) match with k | maxlength

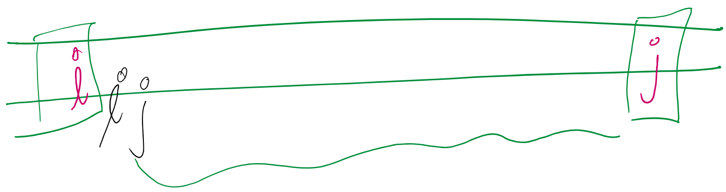
```
for( i=0, i<n; i++)
  for( j=i, j<n; j++)        → sum = 0
  {
    sum += arr[j]
    if (sum == k)
      maxleng = max(leng, j-i+1)
```

(i – j)

Return

→ optimise
→ map?? → Hashmap
→ two pointers?
→ sliding window??

two pointers → Read about it, examples

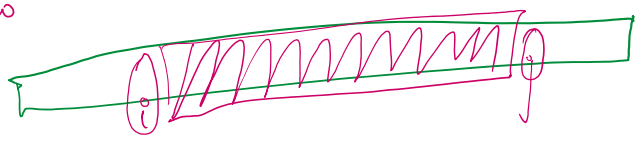


→ left, right
first second

first second

only care about element of those pointers not in between

→ sliding window



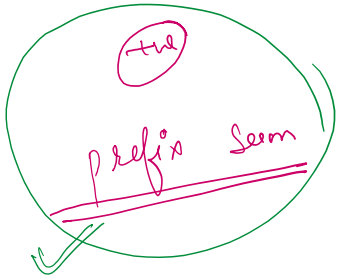
start, end

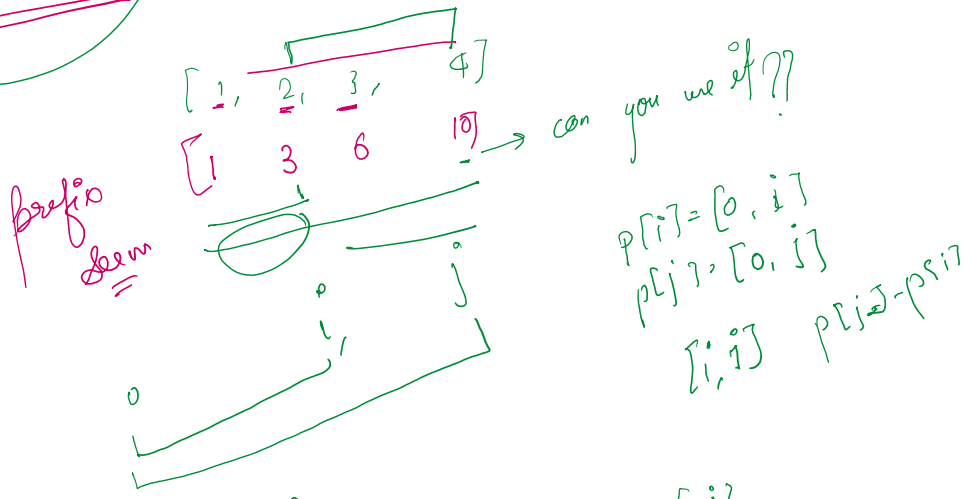
all the elements in the window

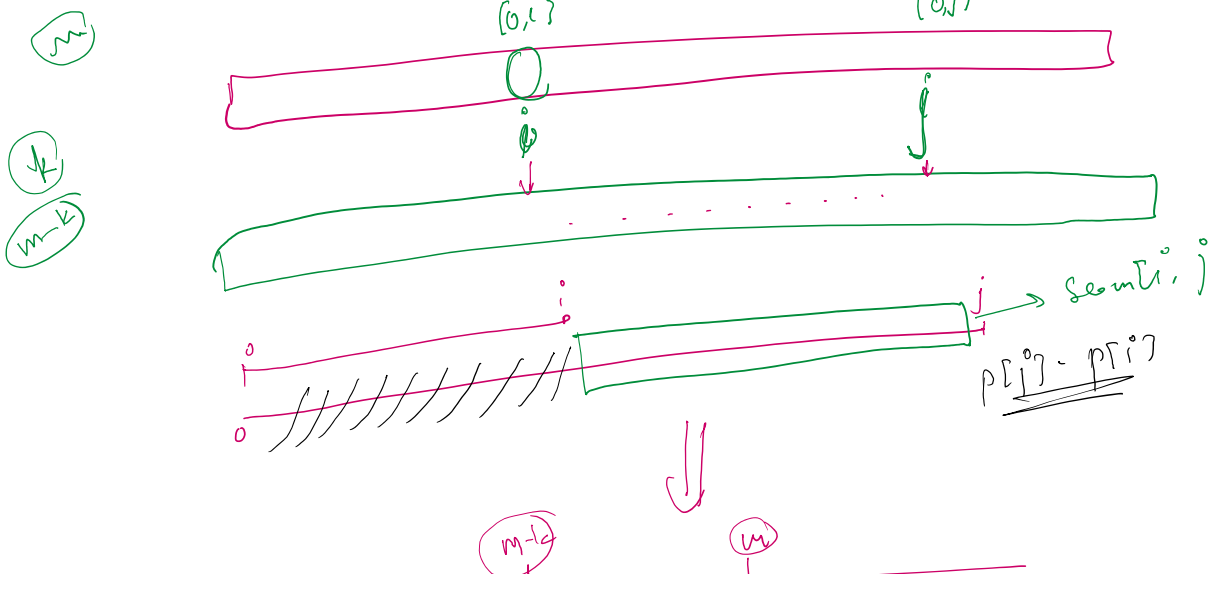
start: only one side

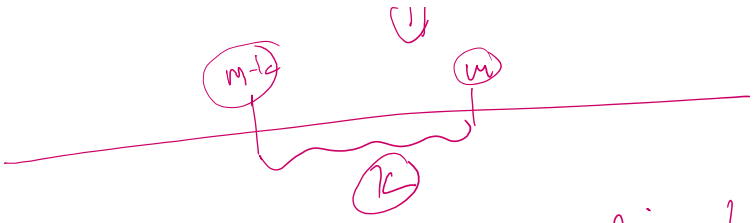
→ fixed window size | (k)

→ variable window size (v)









at least 1 - simple question for techniques