Saturday, 3 July 2021 12:14 PM

Two pointers

Merge Sort

a = [1 3 7 10 12]

b = [1 2 5 6 9]

Merged & sorted array

[1 1 2 3 5 6 7 9 10 12]

$O(n^2) \rightarrow O(n)$

Q

A = [1 3 5 7 15 20]

B = [2 4 5 6 9 10 31]

sorted

C = [1 2 3 3 4 4 6]

< < < < > > > > >

Create another array (C),

C[i] → how many element are there in array 'A' which are smaller than or equal to B[i] ?

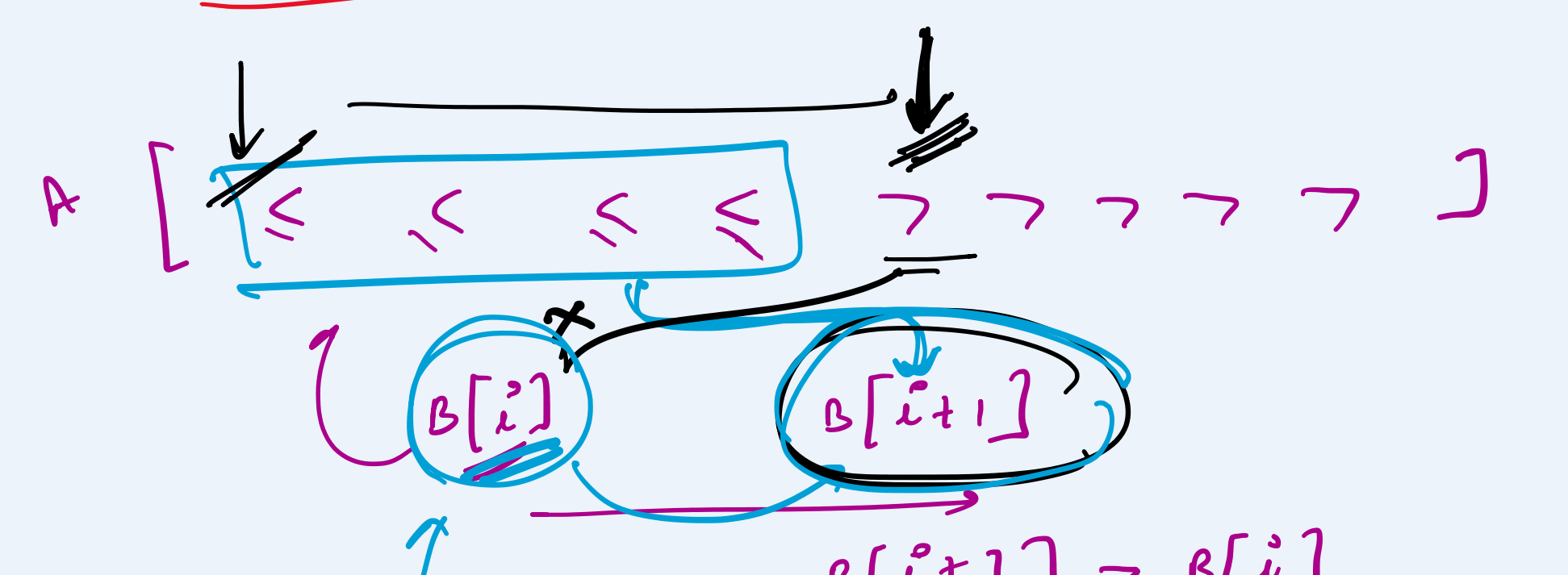
```
{
int j = 0; int cnt = 0;
for (i = 0; i < n; i++) {

    for (   , j < n; j++) {
        if (A[j] <= B[i])
            cnt++;
        else break;
    }
    print(cnt);
}
}
```

$O(n + n)$

↓

$O(2n) \approx O(n)$



B[i+1] > B[i]

i = 1 j = 14

i = 3, j = 12

Q

A = [1, 3, 5, 5, 9, 10, 12, 17] (sorted)

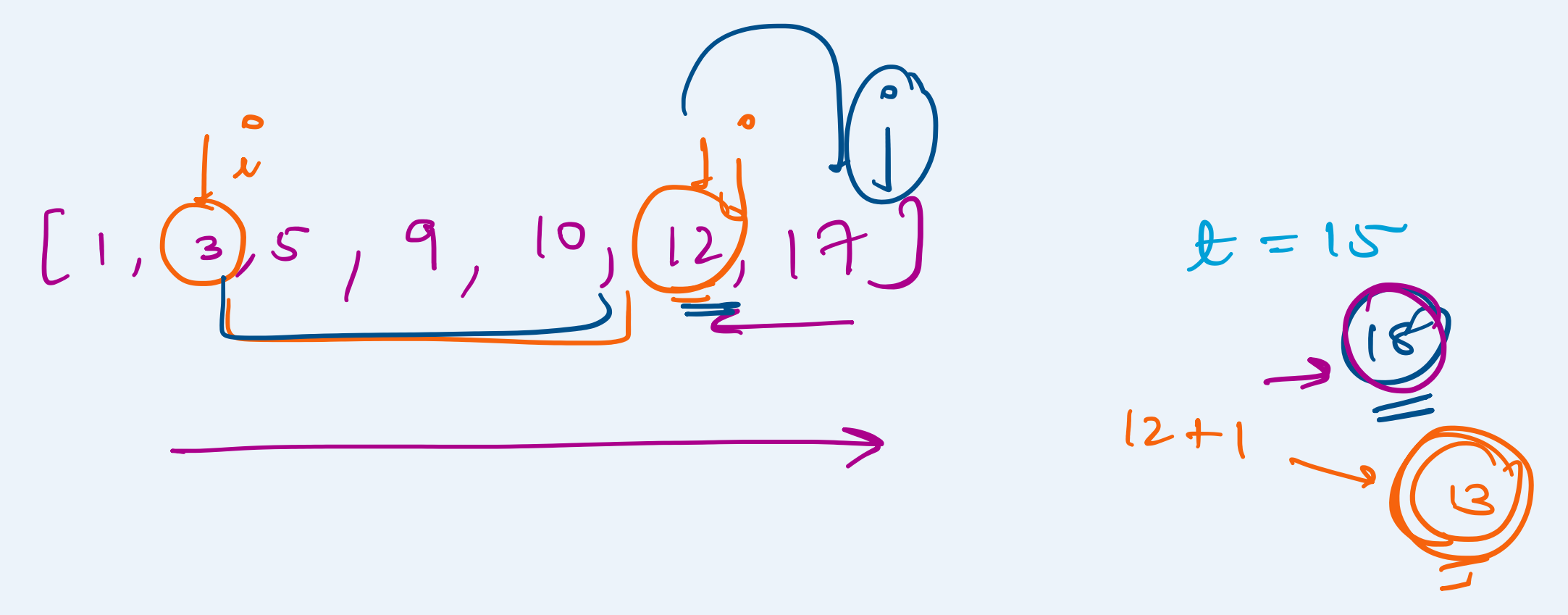
you need to check whether there exists a pair (i, j) such that i ≠ j && A[i] + A[j] = target.

target = 15

```
j = n-1
for (int i = 0; i < n; i++) {
    for ( ; j > i; j-- ) {
        if (a[i] + a[j] == target)
            return true;
        else if (a[i] + a[j] < target) break;
    }
}
```

→ $O(n^2)$

→ $O(n) =$



Q Given a string (s), find the length of longest substring without repeating characters.

s = "abcabcbb" → output → (3)

a
ab
abc
abca
abcabc

s = "bbbb" output → (1)

Create each & every subarray

ans = 0 1 2 3 4 5 (6) (7) sliding window

a b c d e a f b d g c b b

ans = 1

<char, last I + 1>

map →

a	6
b	12
c	11
d	9
e	5
f	7
g	10

ans = 1 2 3 6 (7)

→ (j − i + 1)

(0 − 0 + 1)

i j

ans →

Hashmap → [char, last occur + 1]

```
i = 0;
for j in range(n):
    if str[j] is present in map:
        i = max(map(s[j]), i);
    ans = max(ans, j-i+1);
    map(s[j]) = j+1;
```

Q Given an array, find max length of subarray having sum ≤ k.

arr = [1 3 2 5 1 1 6]

(6) − 1 = (5)

ans → 4

k = 10

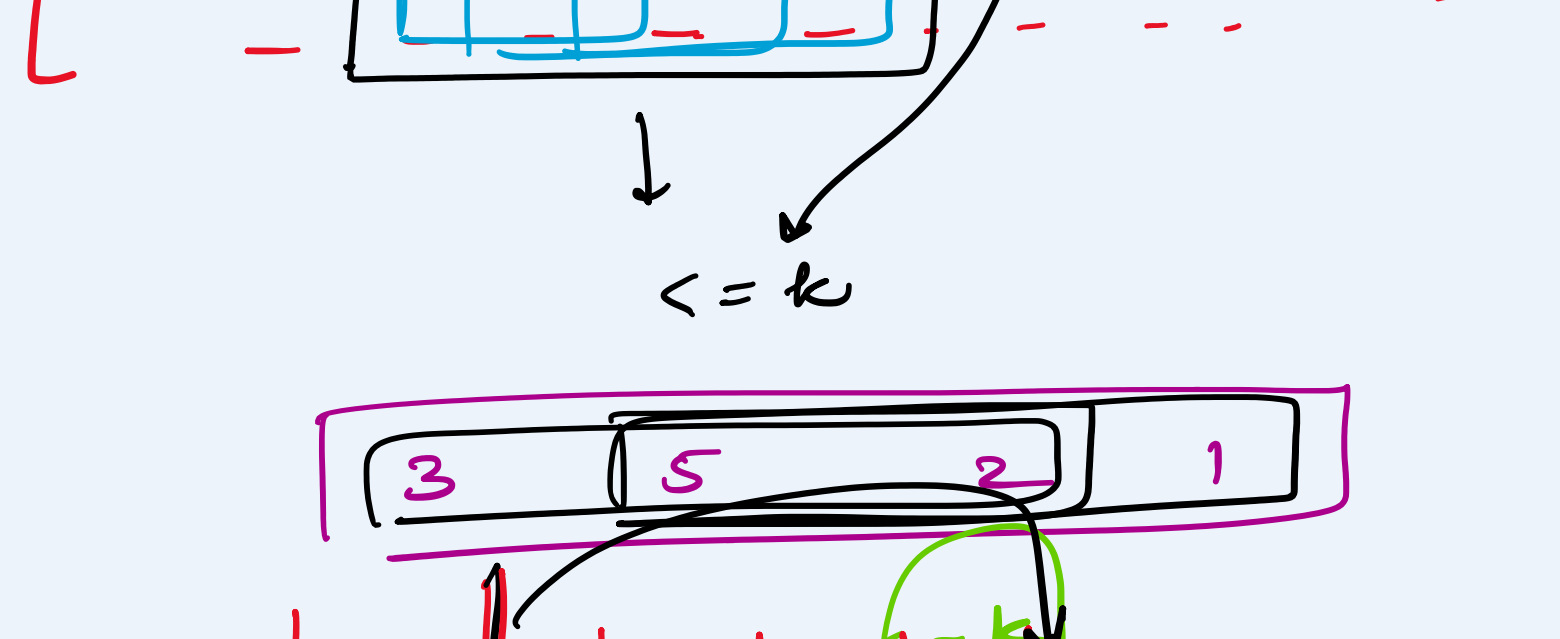
1 + 3 + 2 + (5)

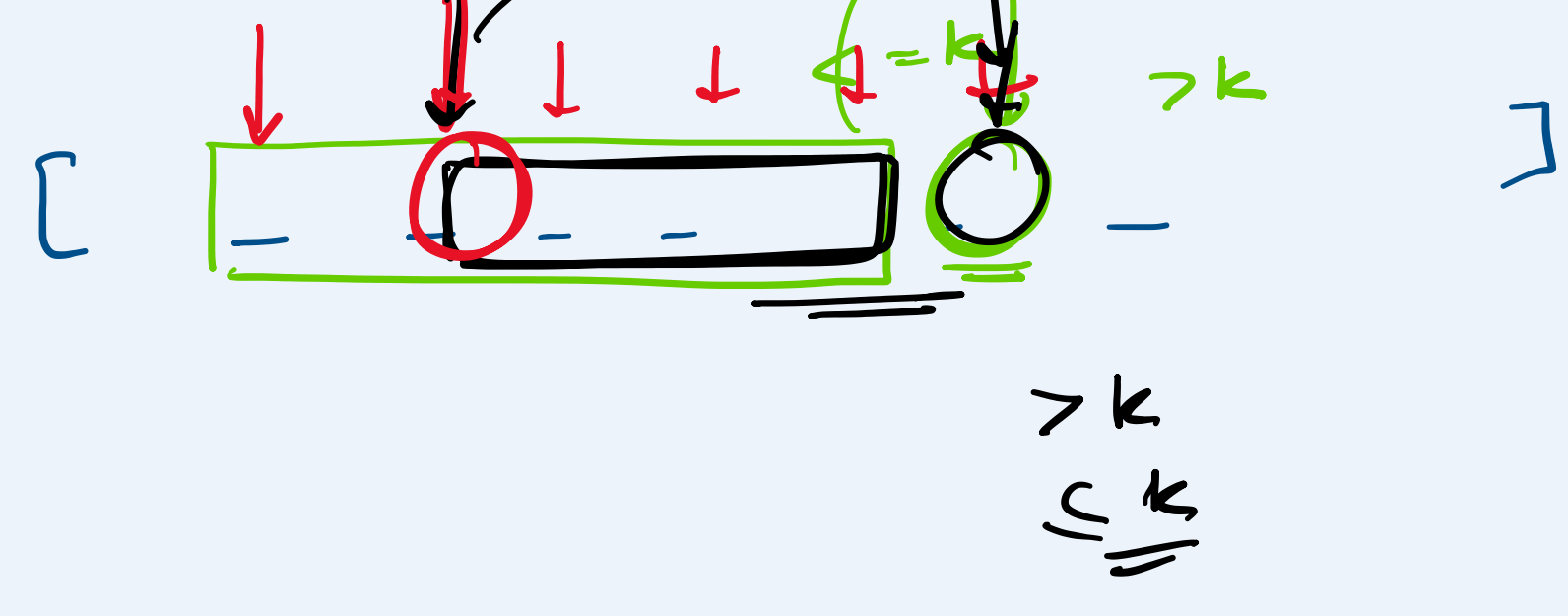
1 + 3 + 2 − 1

+5

$O(n^2) \rightarrow O(n)$

```
ans = 0, j = 0; sum = 0;
for (i = 0; i < n; i++)
    for ( ; j < n; j++) {
        if (sum + a[j] <= k)
            sum += a[j]
    }
    ans = max(ans, j-i);
    sum -= a[i];
}
```





> k

≤ k