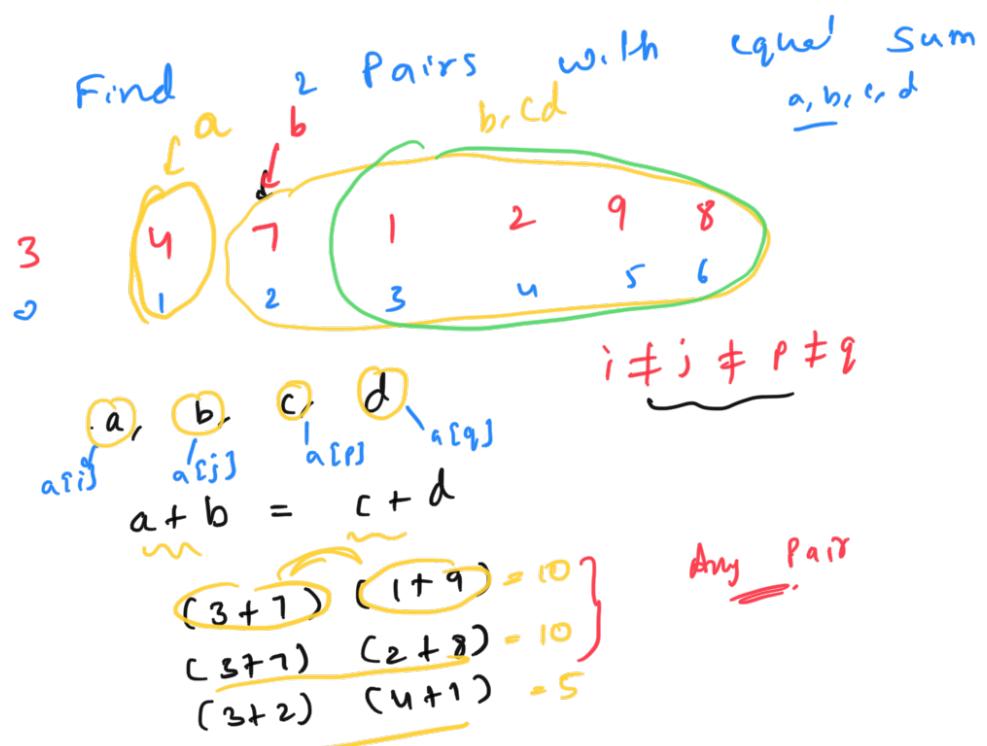


Hashing - 2

Question:

$A =$



Brute Force:

a, b, c, d

↓
4 nested loops

$$a + b = c + d$$

a \leftarrow for(int i = 0; i < n; i++) {
 b \leftarrow for(j = i+1; j < n; j++) {
 c \leftarrow for(p = j+1; p < n; p++) {
 d \leftarrow for(q = p+1; q < n; q++) {
 if($a[i] + a[j] == a[p] + a[q]$)
 return [i, j, p, q]
 }
}
}

T.C: $O(n^4)$

(sort) =
Initialize ?
 $O(1, n-2, n-1)$
 P_1, P_2, P_3, P_4

Efficient Approach:

T.C: $(a+b=k)$

$\Rightarrow O(n)$

a \in $(K-a)$

$$a+b = K$$

$$b = K-a \quad \leftarrow O(1)$$

$\neg K$!
1 chose 2 random number $\boxed{a, b}$ $a+b = c+d$

$$a, b \Rightarrow a+b = c+d$$

$10 = 1, 3, 7, 9$
(create all the pairs?) $(n-1) + (n-2) + (n-3) + \dots + 1$

$$A =$$

$$\begin{matrix} 3 & 4 \\ \diagdown & \diagup \\ (3, 4) & (4, 3) \\ \diagup & \diagdown \\ (4, 7) & (7, 4) \\ \diagdown & \diagup \\ (7, 1) & (1, 7) \\ \diagup & \diagdown \\ (1, 10) & (10, 1) \end{matrix}$$

$$\text{Pairs} =$$

$$\begin{matrix} 1 & 1 \\ \diagdown & \diagup \\ (1, 10) & (10, 1) \\ \diagup & \diagdown \\ (4, 1) & (1, 4) \\ \diagup & \diagdown \\ (4, 9) & (9, 4) \\ \diagup & \diagdown \\ (7, 9) & (9, 7) \end{matrix}$$

$$\begin{matrix} 9 & \\ \diagdown & \diagup \\ (4, 13) & (13, 4) \\ \diagup & \diagdown \\ (3, 1) & (1, 3) \\ \diagup & \diagdown \\ (3, 9) & (9, 3) \end{matrix}$$

T.C: $O(n^2)$

$$\text{No. of Pairs:}$$

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

$$\left[\frac{(n-1)}{2} \right]$$

$$1+2+3+\dots+n-1$$

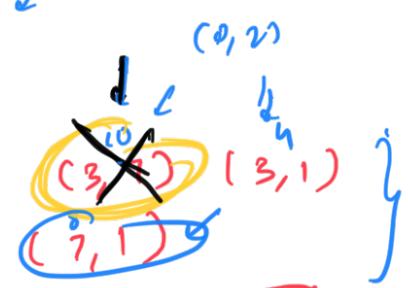
$$\frac{(n-1)(n-1+1)}{2} = \left[\frac{n(n-1)}{2} \right] = \boxed{\binom{n}{2}}$$

HashMap:

$\boxed{1, 3, 7, 9}$
Ex
Assignment



$$\text{array} + \text{array} = \text{sum}$$



sum: $\langle i, j \rangle$

HashMap =

$\begin{cases} 10 : \langle 0, 1 \rangle \\ 4 : \langle 0, 3 \rangle \\ 14 : \langle 1, 2 \rangle \\ 8 : \langle 1, 3 \rangle \end{cases}$

$(1, 3) \quad (0, 3)$

pair < int, int >
vector < >

(struct)

$a, b \in \mathbb{R}^d$

$(i, j) \quad (p, q)$

$i \neq j \neq p \neq q \quad \checkmark$

map<sum, index>

Pairs =

$\begin{cases} 10 : \langle 0, 1 \rangle \\ 4 : \langle 0, 3 \rangle \\ 14 : \langle 1, 2 \rangle \\ 8 : \langle 1, 3 \rangle \end{cases}$

Question:

A:

Subarray

5 . 1 4 6 9

0 1 2 3 4 5 6

Sum

9 3 4

(W)

T.C: $O(n^2) + O(n^2)$
T.C: $O(n^2)$

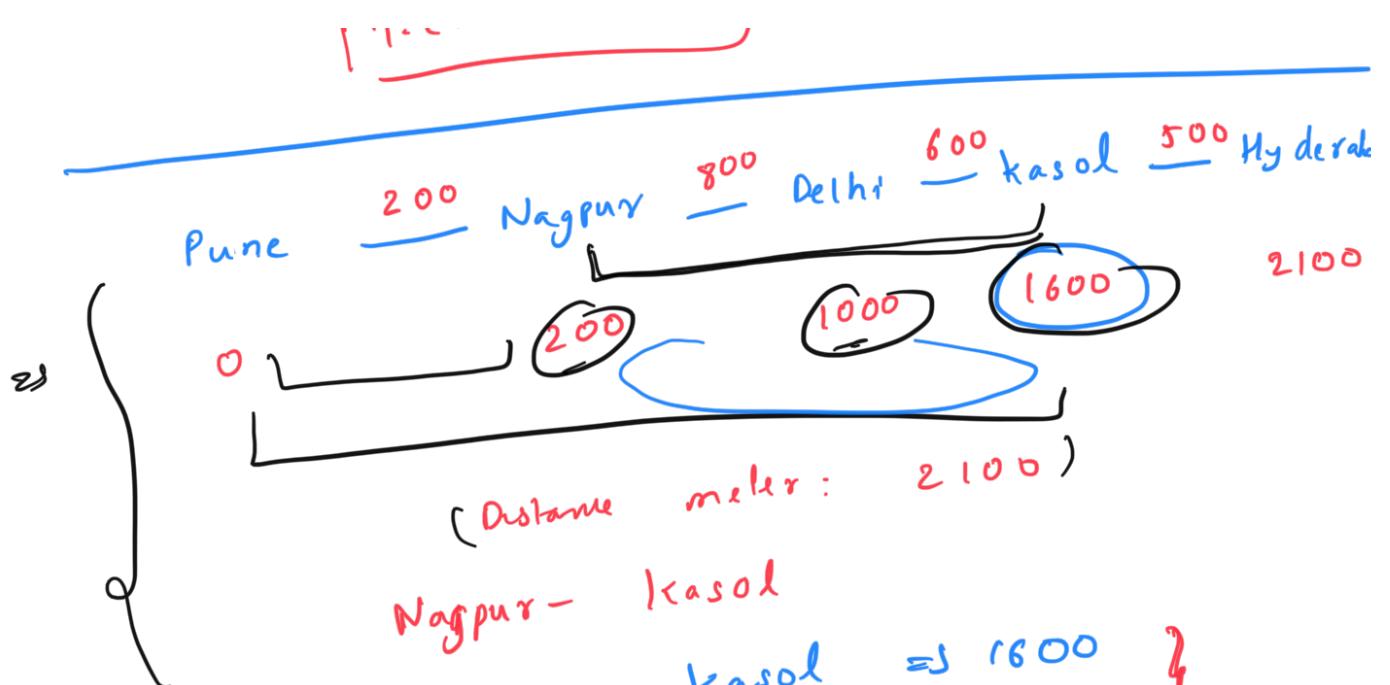
Q queries

i) $(L, R) \Rightarrow [2, 4] \Rightarrow 19$
 $(1, 5) \Rightarrow 23$

Brute Force:

Every query, iterate from
 $L+1$ to R : $O(n \cdot q)$

$[L, R] \Rightarrow O(n)$
 $[0, N-1]$



Nagpur - Kasol

Pune — Nagur \Rightarrow 200
Pune — (L-1) Nagur \Rightarrow 100

A 2

~~Pre~~ ^{sum}

$T-C$:

$$\text{pre}[-1] \quad \underline{\underline{L = 0}}$$

Question :

Subarray with Sum 0

1

$$A = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 1 & -3 & 1 & 2 & -4 & 6 & 2 & -4 & 1 & -5 & 7 \end{bmatrix}$$

— L. Corr.:

卷之三

۱۲۸۷

Bruce Furie

レ^ト

Consider all subarray

$n \Leftarrow [1] [1, -3] [1, -3, 1] \{1, -3, 1, 2\} \dots$

$$n-1 \leftarrow [-3] \quad [-3, 1] \quad \{ -3, 1, 2 \} \quad |-3|, 2, -6?.$$

n-2

11

1

F.C.

$$1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2}$$

$$\frac{n(n+1)}{2} \times n =$$

(L, R)

$O(n^3)$

(L, ρ)

$$\text{Approach 2: } \left(\frac{n(n+1)}{2} \right) \times O(1) = O(n^2)$$

Approach 3:

$A = [0, 1, 2, 3, 4, 6, 2, -4, 1, -5, 7]$

$Pre = [1, -2, -1, 1, -3, 3, 5, 1, 2, -3, 4]$

$pre[i] = \sum(0 \dots i)$

flashset

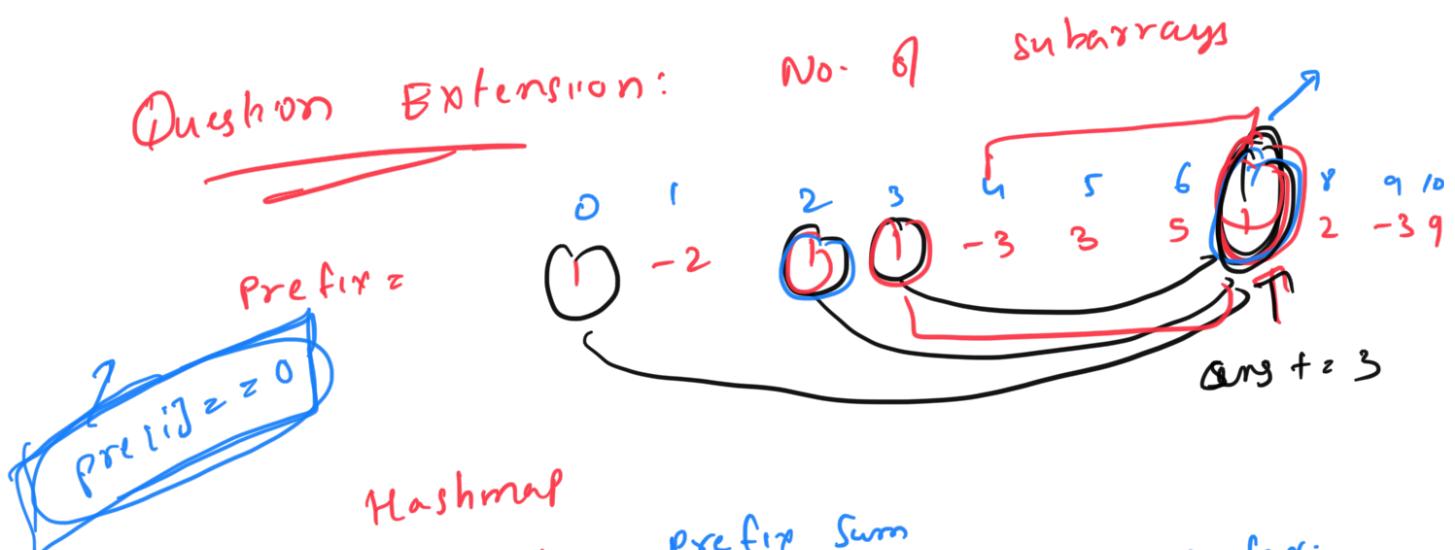
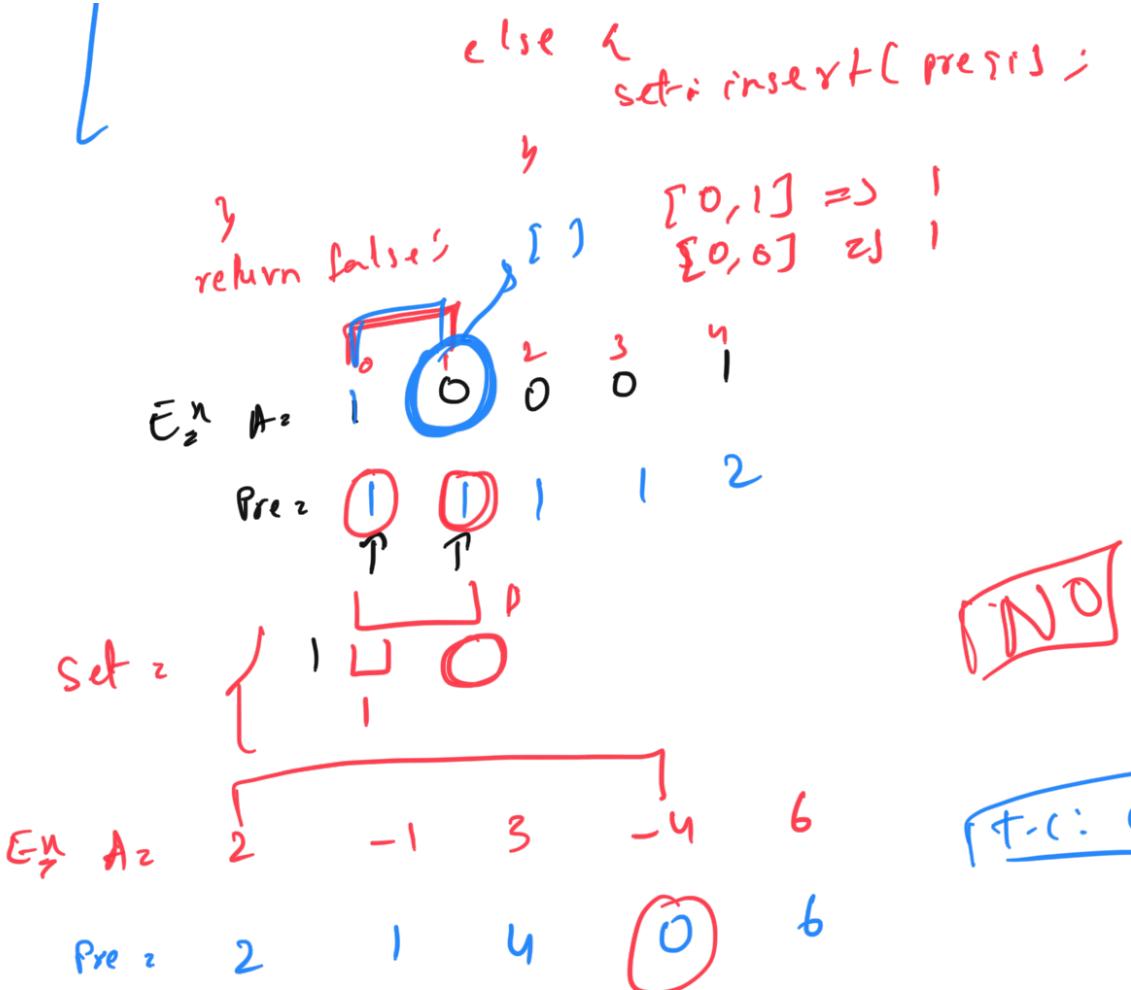


// generate Prefix Arrays ~ $\Rightarrow O(n)$
hashset<int> set;

```
for(int i = 0; i < n; i++) {  
    if(pres[i] == 0) return true;  
}
```

```
    if (pres[i] is in set) &  
        return true;  
}
```

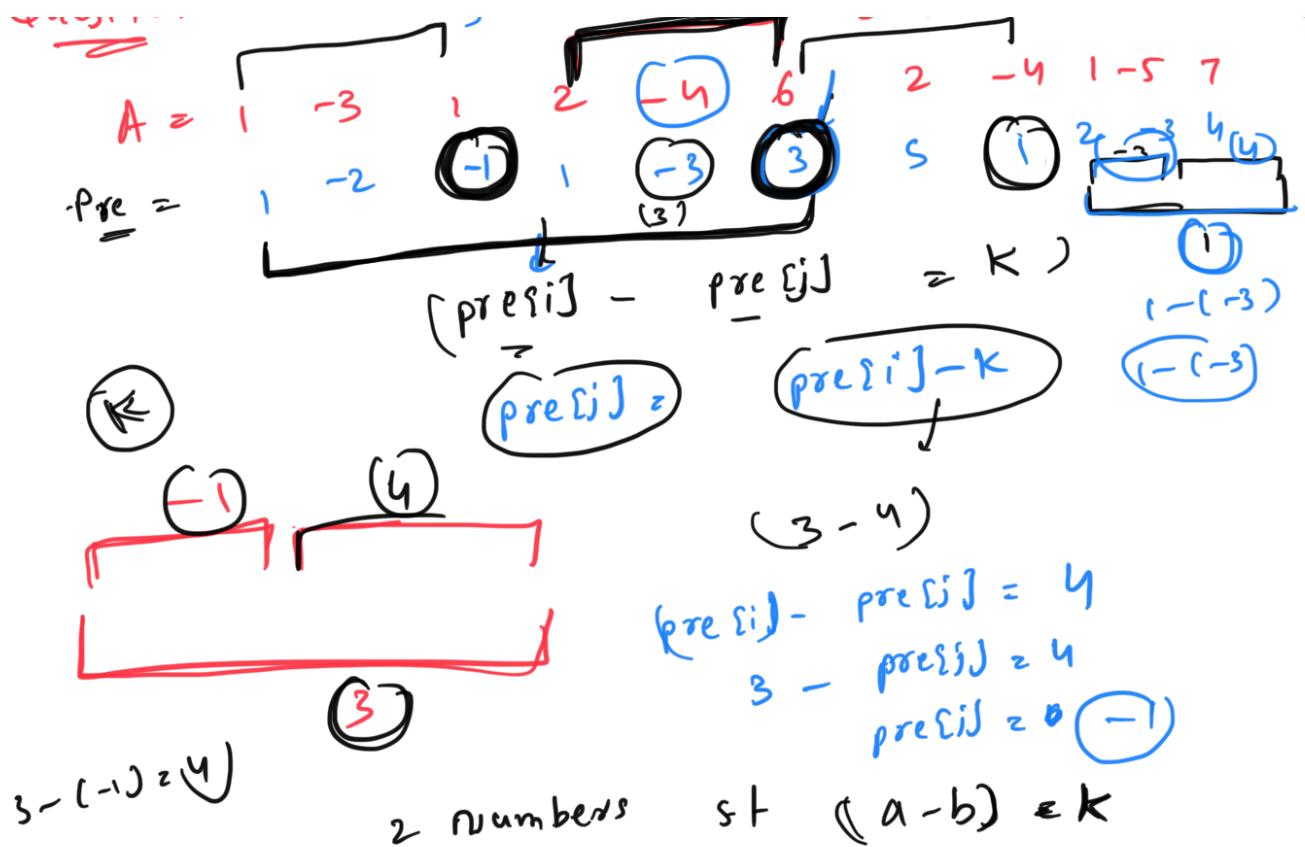
6



Hashmap
 Key: prefix sum
 value: count of that sum so far.

$A = [9, 6, 7, 8]$
 $\text{pre. } [9, 15, 22, 30]$

Duration:
 Subarray → with m sum = K
 $\{6, 2, 1\}$ $K=4$



// Generate Prefix sum

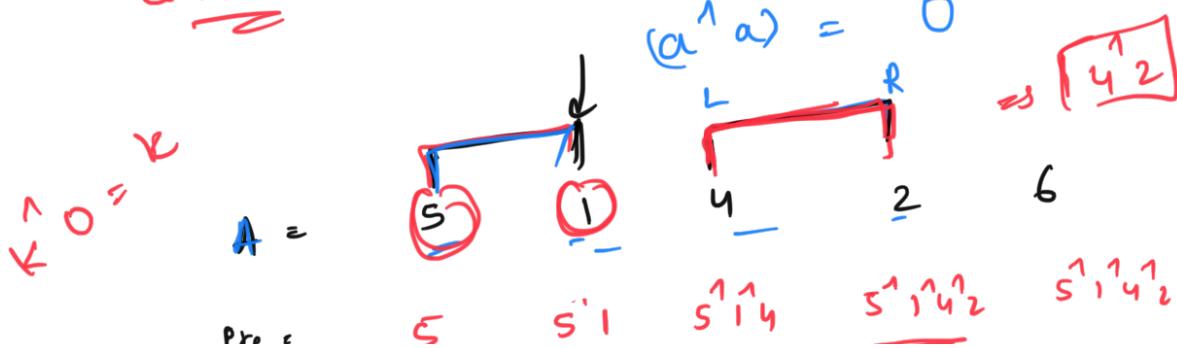
```

    HashSet<int> st;
    for (i = 0; i < n; i++) {
        if (pre[i] == k) return true;
        if (pre[i] - k is in set)
            return true;
        else
            set.insert(pre[i]);
    }

```

Question: Find subarray with $\text{XOR} = k$

$(a^1 a) = 0$



$\text{pre}[i] = \text{xor of all elements from } [0 \dots i]$

$$\begin{array}{r} \text{pre}[5] = 0 \\ \begin{array}{c} (5) \quad (1) \quad (4) \quad (2) \quad (5) \quad (0) \\ \hline (0) \quad (0) \quad (4) \quad (2) \end{array} \end{array} = [4 \ 2]$$

$$\text{pre}[L \dots R] = \text{pre}[R] \wedge \text{pre}[L-1]$$

Question: No. of subarrays with exactly 'K' odd No's
 $[1, 8]$

2 Pointers
 $A = [2, 4, 1, 3, 8]$ $K=1$

$$\{ \begin{array}{l} [2, 4, 1] \quad [4, 1] \quad [1,] \\ [3] \quad [3, 8] \end{array}$$

$$\begin{array}{lll} [2, 4, 1, 3, 8] & [2, 4, 1, 3] & [4, 1, 3] \\ [1, 3, 8] & [1, 3] & [4, 1, 3, 8] \end{array}$$

$K=2$
 $\text{sum} \geq K$
 $\text{Replace } \begin{cases} \text{Even} \\ \text{Odd} \end{cases} \quad \begin{cases} \text{Number with 0} \\ \text{Number with 1} \end{cases}$ $\begin{cases} \text{sum} = 3 \\ 3 \text{ ones} \end{cases}$

$$A = [0, [0, 1, 1], 0]$$

No. of subarrays with sum = K

$$n, n+1, n+2, n+3 \rightarrow 4, 5, 6$$

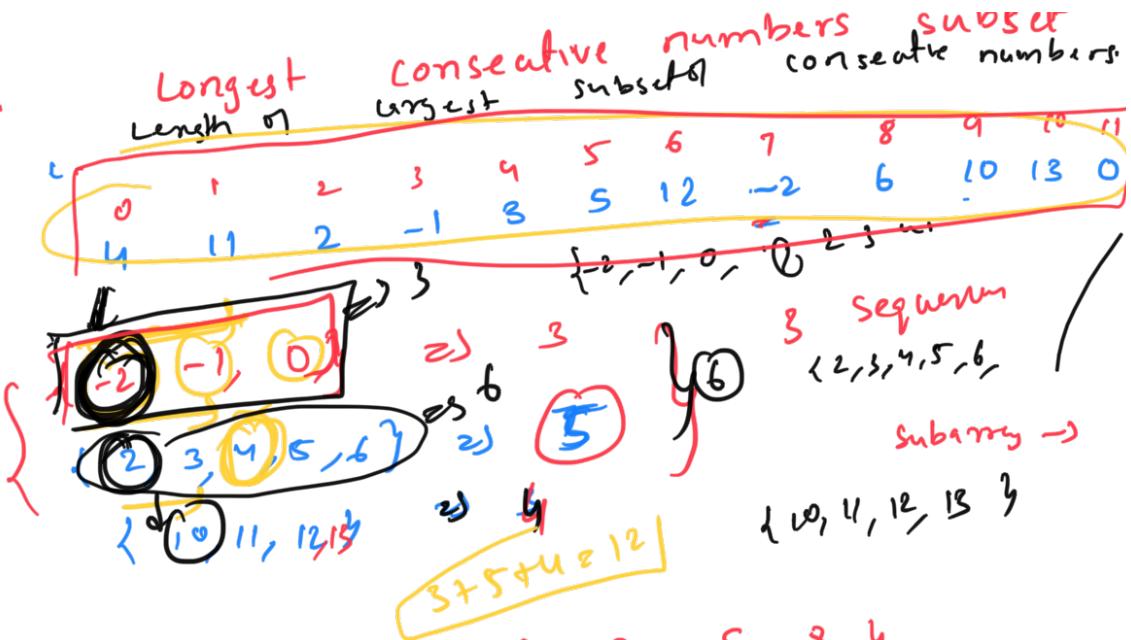
$$[0, 0, 0, 0, 0]$$

... 1

Question:

$$A = \{4, 11, 2, -1, 3, 5, 12, -2, 6, 10, 13, 0\}$$

1st class



contiguous?

(Ex)

$$\{7, 4, 2, 6, 3, 5, 8\} \rightarrow [7]$$

{2, 3, 4, 5, 6, 7, 8} → [7]

Approach 1:

$O(n \log n)$

sort

E_1^n

sort

E_2^n

$$\text{len} = \emptyset \neq 2 \neq 4$$

$$\text{len} = \emptyset \neq 2 \neq 4$$

1. -

2. -

3. -

4. -

5. -

6. -

7. -

8. -

9. -

10. -

11. -

12. -

13. -

14. -

15. -

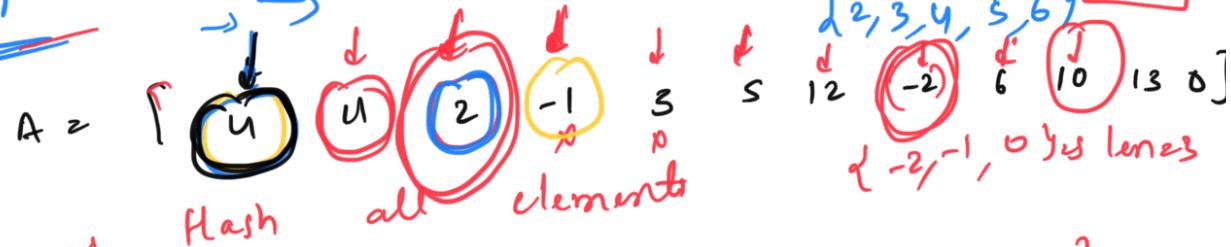
ans = 2' 5

(Advanced)
Merge
Heap
Queue

T.C: $O(n \log n) + O(n)$

($-O(1)$) T.C: $O(n \log n + n) \Rightarrow O(n \log n)$

Approach 2:



set: {4, 11, 2, -1, 3, 5, 12, -2, 6, 10, 13, 0}

{10, 11, 12, 13} → len = 4

ans = 2' 5

... {4, 5, 6} → 3

$O(n^2)$

$\forall i: \{11, 12, 13\} \Rightarrow 3$
 $\exists j: \{2, 3, 4, 5, 6\} \Rightarrow 5$

$$T.C = \frac{n \times n}{2} = O(n^2)$$

$$A = \begin{matrix} & & & & & \\ \downarrow & \downarrow & \downarrow & & & \\ 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \quad (n)$$

$$(n) + (n-1) + (n-2) + \dots + 1 = \boxed{O(n^2)}$$

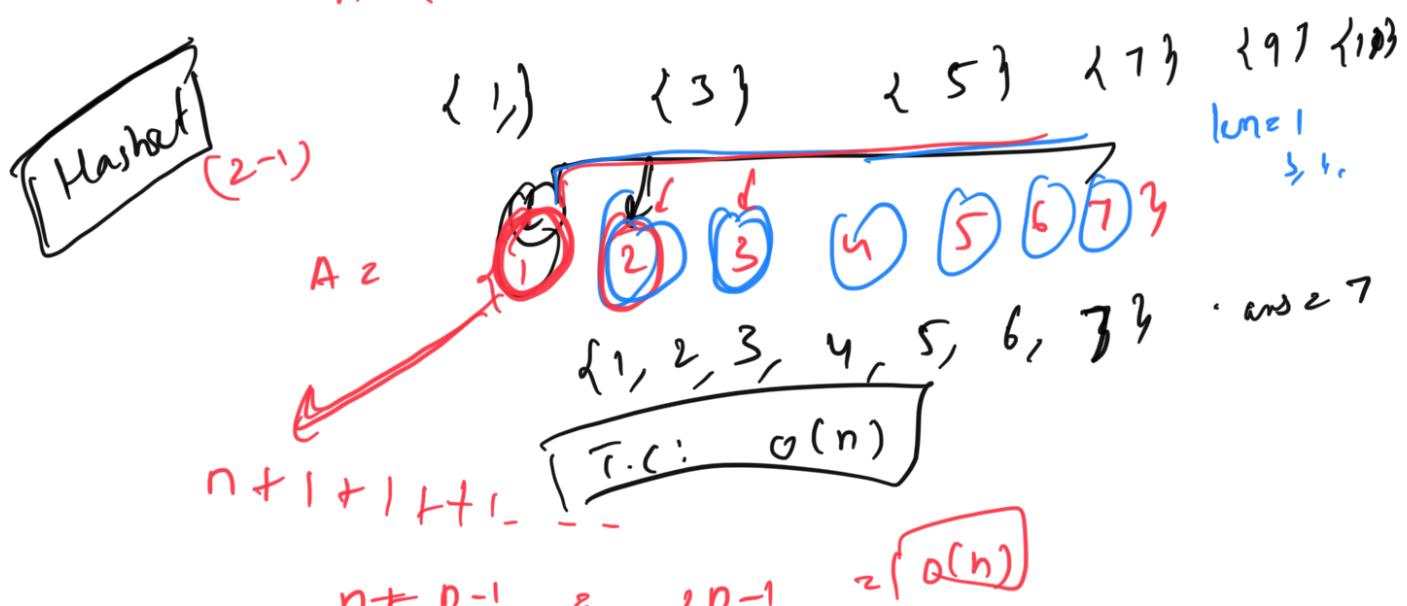
Approach 3:

$(a[i:j])$ exists in hashset or not? $\downarrow O(1)$
 $n \times O(1) \Rightarrow \boxed{O(n)}$

k sequences \Rightarrow we are expanding on k elements
 In total, we'll visit exactly n elements $\Rightarrow O(n)$

$\{1, 2, 3, 5\}$

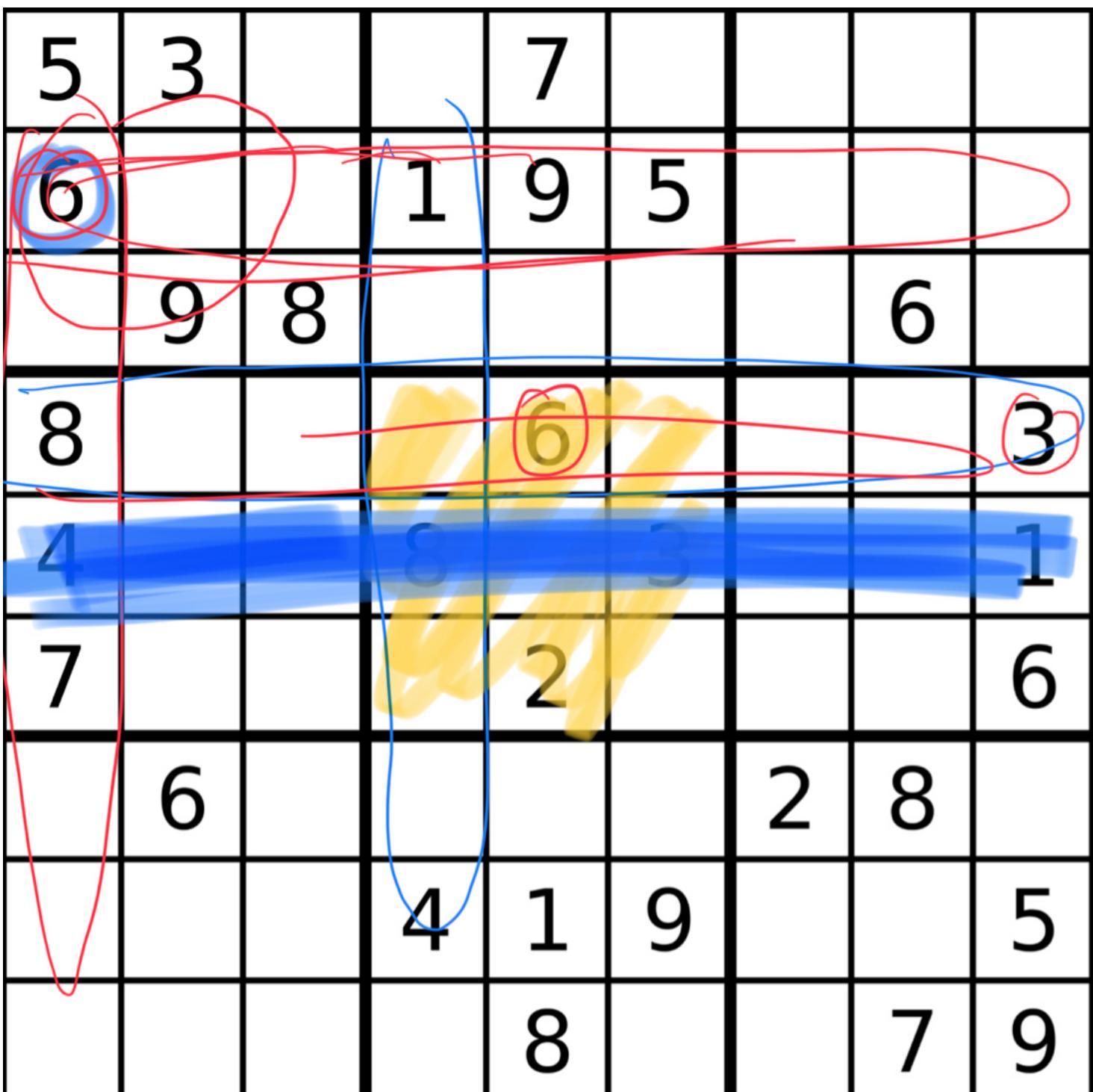
$K = n$
 $A = \{1, 3, 5, 7, 9, 11\}$



Pseudocode :
~~//~~ ans = INT-MIN;
 Hash the array & count a set }
 for (int i=0; i<n; i++) {
 if (arr[i]-1 is in set) {
 continue;
 }
 else {
 len++;
 for (j = arr[i]+1; j++ < INT-MAX) {
 if (j is in set) len++;
 else {
 ans = max (ans, len);
 break;
 }
 }

INT-MIN
 arr[i]-1
 ↓
 arr[i] > INT-MAX
 long long

Sudoku :



Every row should all number (1-9)
 Every column (1-9)

Brute Force

[element: $O(n) + O(n) + O(n) \approx O(n)$

Total cells: n^2

T.C: $O(n^3)$

$$n = 9$$

$$\frac{81}{n^3}$$

$\boxed{O(n^3)}$

T.C: $\boxed{O(n^3)}$
 $O(9^3)$

$$(10^6 - 10^7)$$

1 row $\approx O(n)$

$$O(n^2)$$

$$O(n^3)$$

$$O(n^4)$$

$$O(n^5)$$

(9) $\frac{n}{n}$ rows \approx
 (9) n columns \approx
 (9) n boxes \approx

27 Hash sets? $O(n^3) + O(n^2) + O(n) \approx O(n^4)$

$$(n = 9)$$

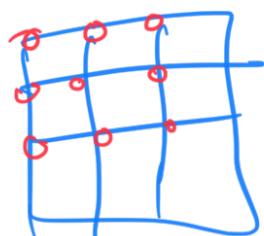
$$(81)$$

How many hash sets?

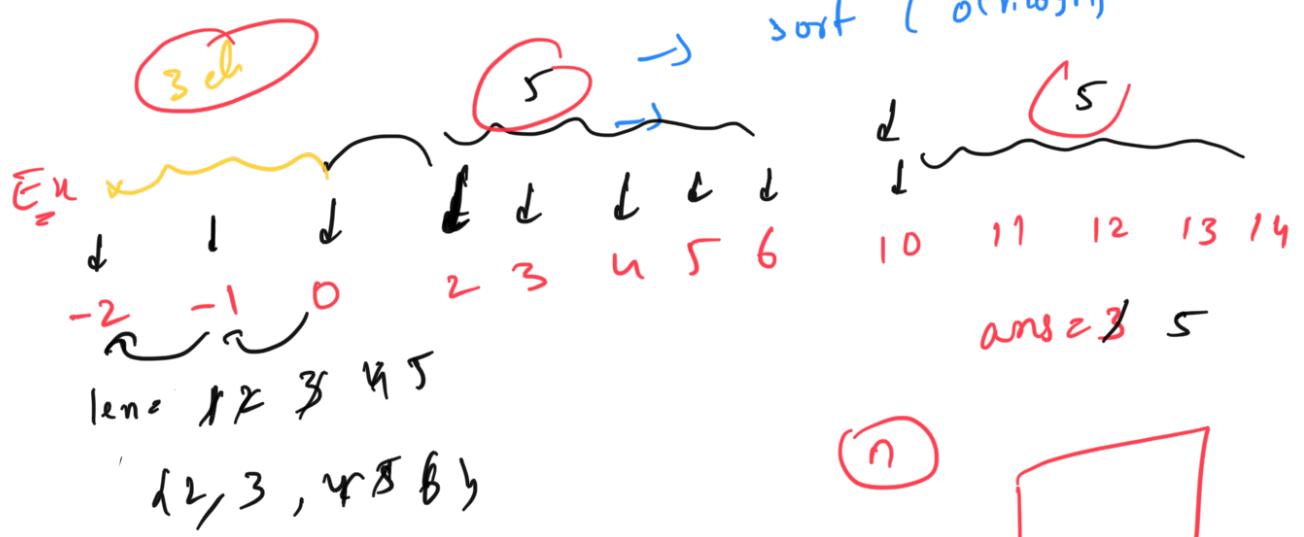
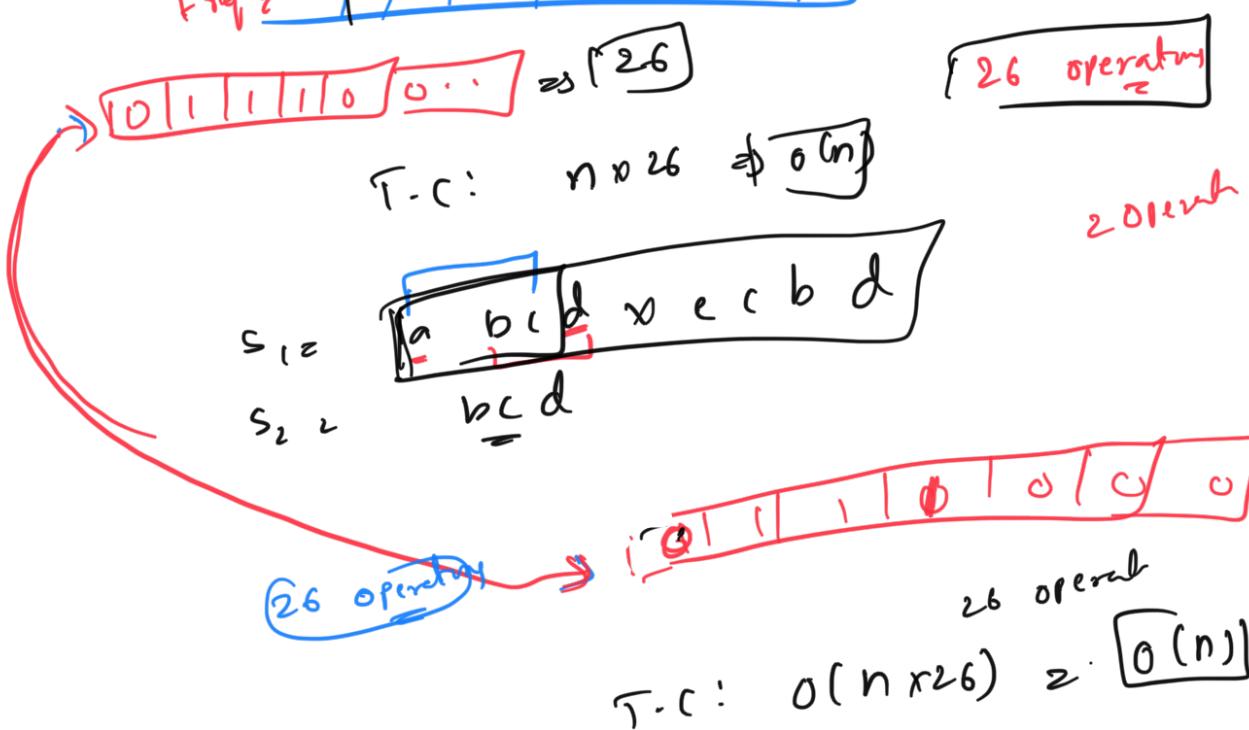
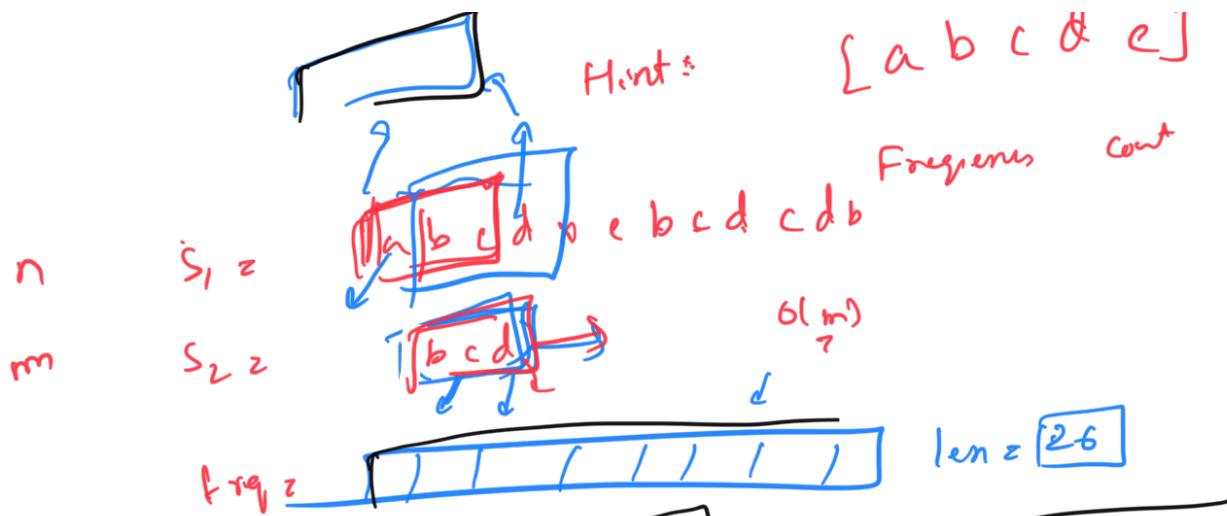
Reuse a hashset

$$\approx 3 + n$$

$\boxed{1 \text{ hashset}}$



$$\boxed{i + j + 1}$$



$O(n^2)$

Lo.

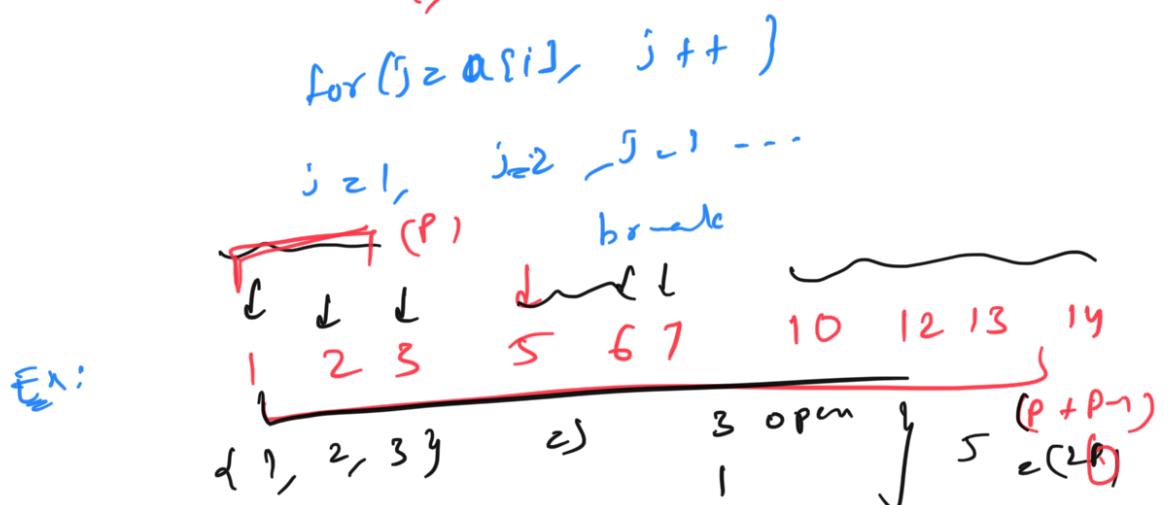
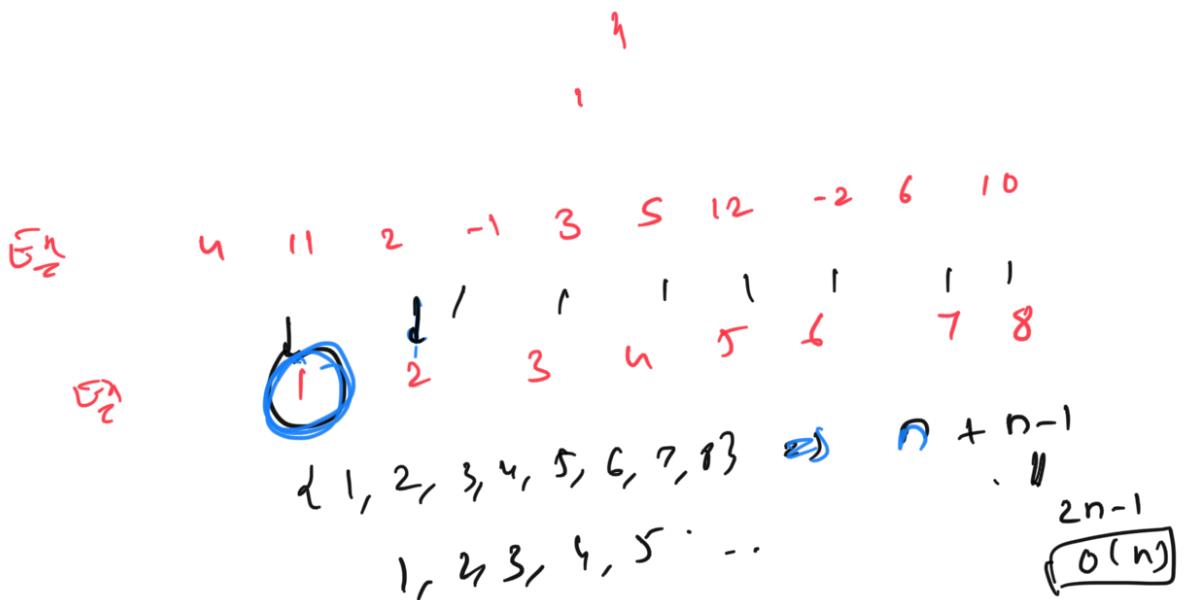
{ for () {

do T.C: $O(n^2)$

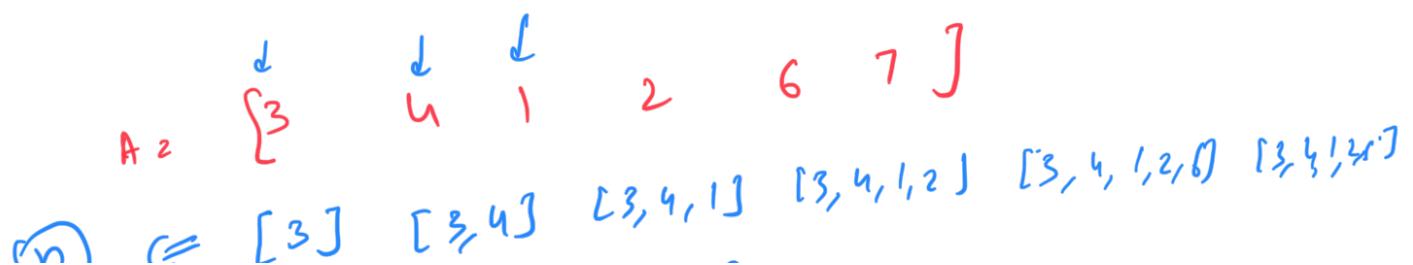
(n)



{ for {



$$2(p+q+r) \leq 2n \Rightarrow O(n)$$



$$\begin{array}{ccccccc}
 \text{Sum} & = & [u] & [u, 1] & [u, 12] & \dots & - \\
 (n-1) & & & & & & \\
 n-2 & & [1] & & & & \\
 \vdots & & & & & & \\
 1 & & & & & & \\
 \end{array}$$

$$1+2+3+\dots+n \quad \text{with } \boxed{\frac{n(n+1)}{2}}$$

$$\begin{array}{ccccccccc}
 A = & 1 & -3 & 1 & 2 & -4 & 6 & 2 & -4 & 1 & -5 & 7 \\
 \text{pre} = & 5 & -2 & -1 & 5 & -3 & 3 & 5 & 1 & 2 & -3 & 4
 \end{array}$$

$$\text{set} = \{ 5, -2, -1,$$

Zero Sum subarray

```

bool checkZeroSubarray(int pre[], int n){
    unordered_set<int> st;
    for(int i=0;i<n;i++){
        if(pre[i] == 0)
            return true;
        if(st.find(pre[i]) != st.end())
            return true;
        else
            st.insert(pre[i]);
    }
    return false;
}

```

Array with sum K

```
bool checkKSubarray(int pre[], int n, int k){  
    unordered_set<int> st;  
    for(int i=0;i<n;i++){  
        if(pre[i] == k)  
            return true;  
        if(st.find(pre[i] - k) != st.end())  
            return true;  
        else  
            st.insert(pre[i]);  
    }  
    return false;  
}
```

Longest subset with consecutive elements

```
for(int i= 0 ; i<n; i++){  
    if(st.find(a[i] - 1) != st.end())  
        continue;  
    else{  
        int len = 1;  
        for(j = a[i] + 1; ;j++){  
            if(st.find(j) != st.end()) len++  
            else{  
                ans = max(ans, len);  
                break;  
            }  
        }  
    }  
}
```

Count no. of subarrays with 0 sum

```
bool checkZeroSubarray(int pre[], int n){  
    unordered_map<int, int> mp;  
    ans = 0;  
    for(int i=0;i<n;i++){  
        if(pre[i] == 0)  
            ans++;  
        if(mp.find(pre[i]) != mp.end())  
            ans = ans + mp[pre[i]];  
        mp[pre[i]]++;  
        else  
            mp.insert({pre[i], 1});  
    }  
    return false;  
}
```