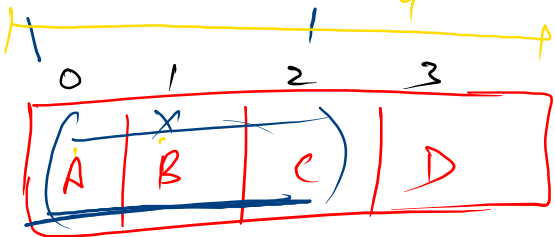
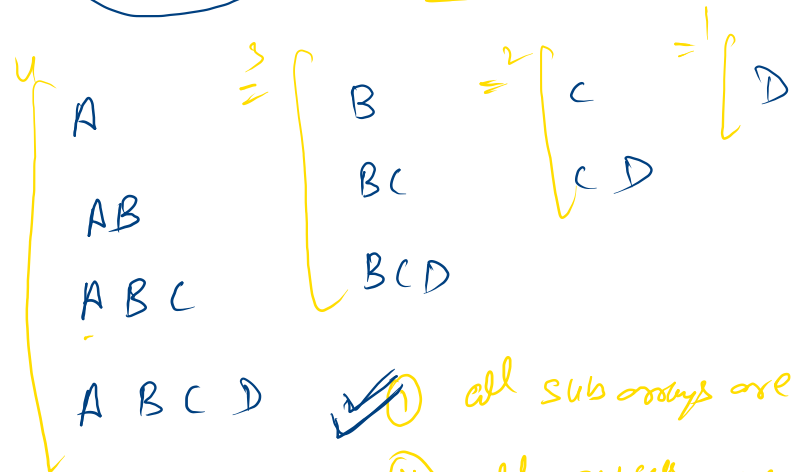


- Subarray / Subset Concept
- Subarray Code
- Binary Search → Concept
- Questions
-

AC



Sub-array  $\Rightarrow$  Contiguous



- ✓ (i) all subarrays are subsets
- (ii) all subsets are subarrays

no. elements  $\Rightarrow$   $n(\text{Subarray}) = \frac{n(n+1)}{2}$

$n + (n-1) + (n-2) + \dots + 1 \Rightarrow \frac{n(n+1)}{2}$

Subset  $\Rightarrow$  Non-Contiguous

0/1  
exc inc

| A | B | C | D |             |
|---|---|---|---|-------------|
| ↑ | ↑ | ↑ | ↑ |             |
| 0 | 0 | 0 | 0 | = -         |
| 0 | 0 | 0 | 1 | = D         |
| 0 | 0 | 1 | 0 | = C         |
| 0 | 0 | 1 | 1 | = CD        |
| 0 | 1 | 0 | 0 | = B         |
| 0 | 1 | 0 | 1 | = <u>BD</u> |
| 0 | 1 | 1 | 0 | = BC        |
| 0 | 1 | 1 | 1 | = BCD       |
| 1 | 0 | 0 | 0 | = A         |
| 1 | 0 | 0 | 1 | = AD        |
| 1 | 0 | 1 | 0 | = AC        |
| 1 | 0 | 1 | 1 | = ACD       |

| A | B | C | D |        |
|---|---|---|---|--------|
| ↑ | ↑ | ↑ | ↑ |        |
| 1 | 1 | 0 | 0 | = AB   |
| 1 | 1 | 0 | 1 | = ABD  |
| 1 | 1 | 1 | 0 | = ABC  |
| 1 | 1 | 1 | 1 | = ABCD |

$\frac{2}{A} \cdot \frac{2}{B} \cdot \frac{2}{C} \cdot \frac{2}{D} \Rightarrow 2^4$

no. elements  $\Rightarrow 2^n$

no. of  
Subsets

$\rightarrow$  no. of Subarray

10

20

30

$(0,0) \leftarrow 10$  ✓

$(0,1) \quad 10 \quad 20$  ✓

$(0,2) \quad 10 \quad 20 \quad 30$  ✓

$(1,1) \quad 20$  ✓

$(1,2) \quad 20 \quad 30$  ✓

$(2,2) \quad 30$  ✓

arr →

| 0  | 1  | 2  |
|----|----|----|
| 10 | 20 | 30 |

| 0  | 1  | 2  | 3  |
|----|----|----|----|
| 10 | 20 | 30 | 40 |

$\uparrow$   $\uparrow$   
 $i$   $j$

10

10 20

10 20 30

10 20 30 40

for ( i = 0 ; i < n ; i++ ) {

for ( j = i ; j < n ; j++ ) {

}

)

|    |    |    |
|----|----|----|
| 10 | 20 | 30 |
|----|----|----|

0

1

2

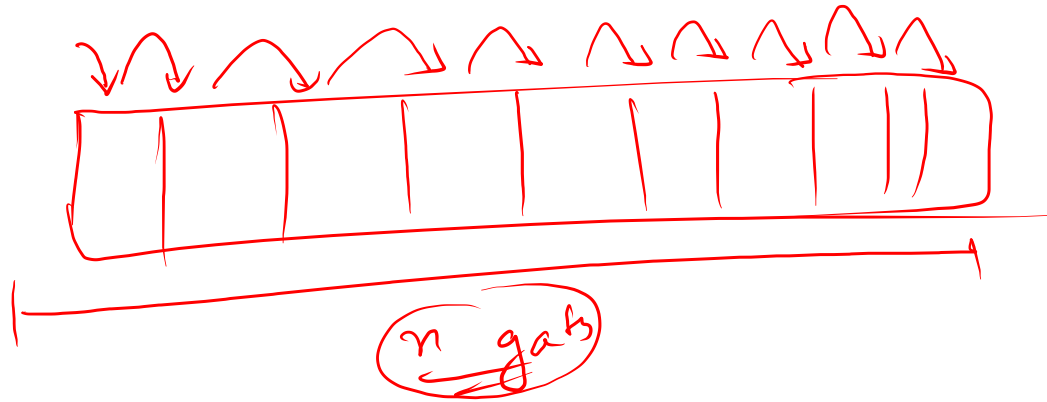
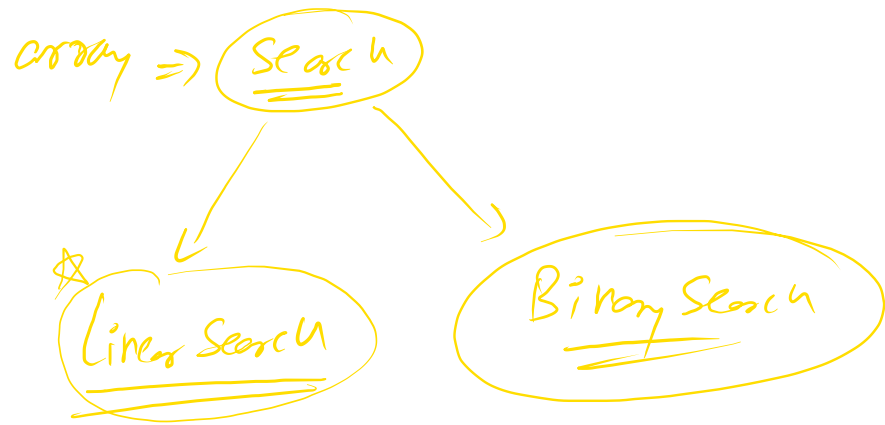


idx = 0 / 1 / 2 / 3

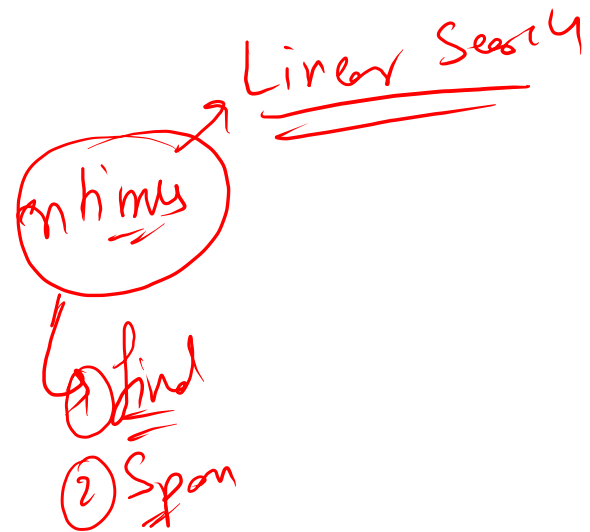
10 - 20 - 30 -



```
public static void subArrayProblem(int[] arr){
    for(int i = 0 ; i < arr.length ; i++){
        for(int j = i ; j < arr.length ; j++){
            for(int idx = i ; idx <= j ; idx++){
                System.out.print(arr[idx]+"\\t");
            }
            System.out.println();
        }
    }
}
```



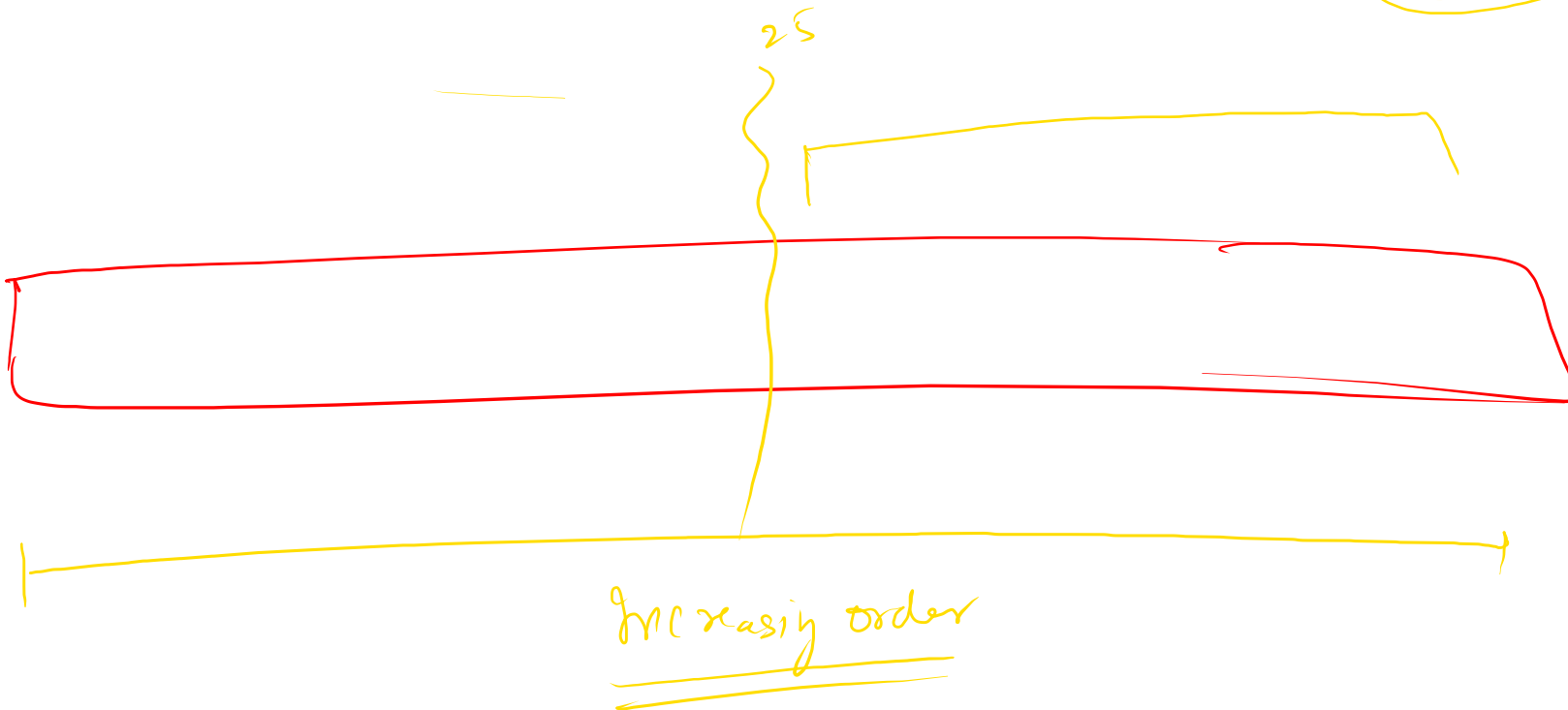
$x = 30$



Binary Search

→ Sorted array

K = 30



k=23

|    |    |          |          |          |    |    |    |    |    |    |    |     |
|----|----|----------|----------|----------|----|----|----|----|----|----|----|-----|
| 0  | 1  | <u>2</u> | <u>3</u> | <u>4</u> | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12  |
| 10 | 15 | 18       | 25       | 36       | 42 | 43 | 50 | 89 | 90 | 95 | 99 | 100 |

↑    ↑  
ed   st

(st, ed) ⇒ find?

$$\text{mid} = (st + ed) / 2$$

=

element doesn't exist

$$(0, 12) \rightarrow 6$$

$$(0, 5) \rightarrow 2$$

$$(3, 5) \rightarrow 4$$

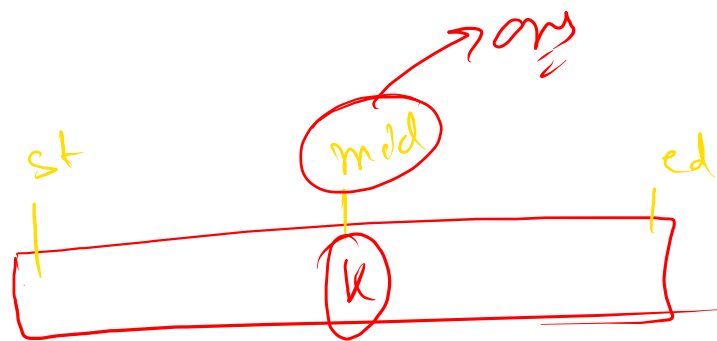
$$(3, 3) \rightarrow 3$$

$$(3, 2) \rightarrow \text{stop}$$

(A, B)    ✓ A > B  
              ✓ A < B  
              A == B

```
if (arr[mid] < k) {  
    st = mid + 1;  
}  
else if (arr[mid] > k) {  
    ed = mid - 1; ✓  
}  
else {  
    element found;  
}
```





|    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|
| 0  | 1  | 2  | 3  | 4  | 5  | 6  |
| 10 | 13 | 15 | 18 | 19 | 20 | 25 |

$k = 21$

```

public static int binarySearch(int arr[], int k){
    int st = 0, ed = arr.length-1;

    while(st <= ed){
        int mid = (st + ed)/2;

        if(arr[mid] < k){
            st = mid+1;
        } else if(arr[mid] > k){
            ed = mid-1;
        } else{
            return mid;
        }
    }

    return -1;
}

```



In a country of novice government, the economic system is changed where only coins are used that too of various denominations. Whenever a foreigner visits this country, they visit a money exchanger to get the currency of the same country. As the foreigner is unaware of the denomination of the country, the money exchange prefers to tell them the denomination which is the nearest maximum and nearest minimum to the denomination mentioned by the foreigner. In case they get the correct guess of the denomination, they are told the same denomination. The denominations are always quoted in ascending order.

Example 1: In a country, 8 given denominations are as follows

→ [5, 10, 15, 22, 33, 40, 42, 55]

The foreigner asks for denomination 25.

The money exchange tells them that denominations of 33 and 22 are available.

ceil & floor

Example 2:

In a country, 5 given denominations are as follows

→ [7, 14, 18, 25, 30]

The foreigner asks for the denomination of 18.

The money exchange tells them a denomination of 18 is available.

ceil → just larger ] element doesn't exist

floor → just smaller

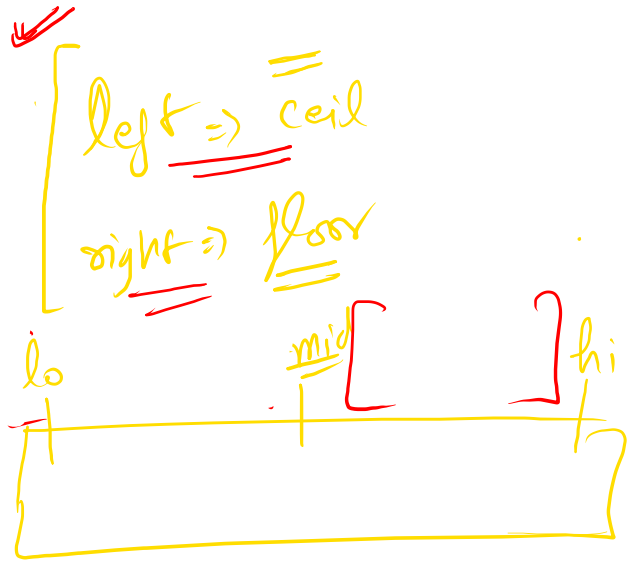
element exist → element

(K = 34)

arr

=

|   |   |    |    |    |    |    |    |    |    |
|---|---|----|----|----|----|----|----|----|----|
| 0 | 1 | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  |
| 1 | 5 | 10 | 15 | 22 | 33 | 40 | 42 | 55 | 66 |



| lo | hi | mid |
|----|----|-----|
| 0  | 9  | 4   |
| 5  | 9  | 7   |
| 5  | 6  | 5   |
| 6  | 6  |     |

ceil  $\Rightarrow$  just larger

floor  $\Rightarrow$  just smaller

✓ ceil = ~~1~~ ~~7~~ 6

✓ floor = ~~7~~ ~~4~~ 5



first idx

K = 33

→ repeat

, sorted

|   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 | 1 | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| 1 | 5 | 10 | 15 | 22 | 33 | 33 | 33 | 33 | 33 | 33 | 40 | 42 | 55 | 66 | 77 |

↑   ↑  
hi   lo

first idx = 5

[ better first occurrence → left

[ better last occurrence → right