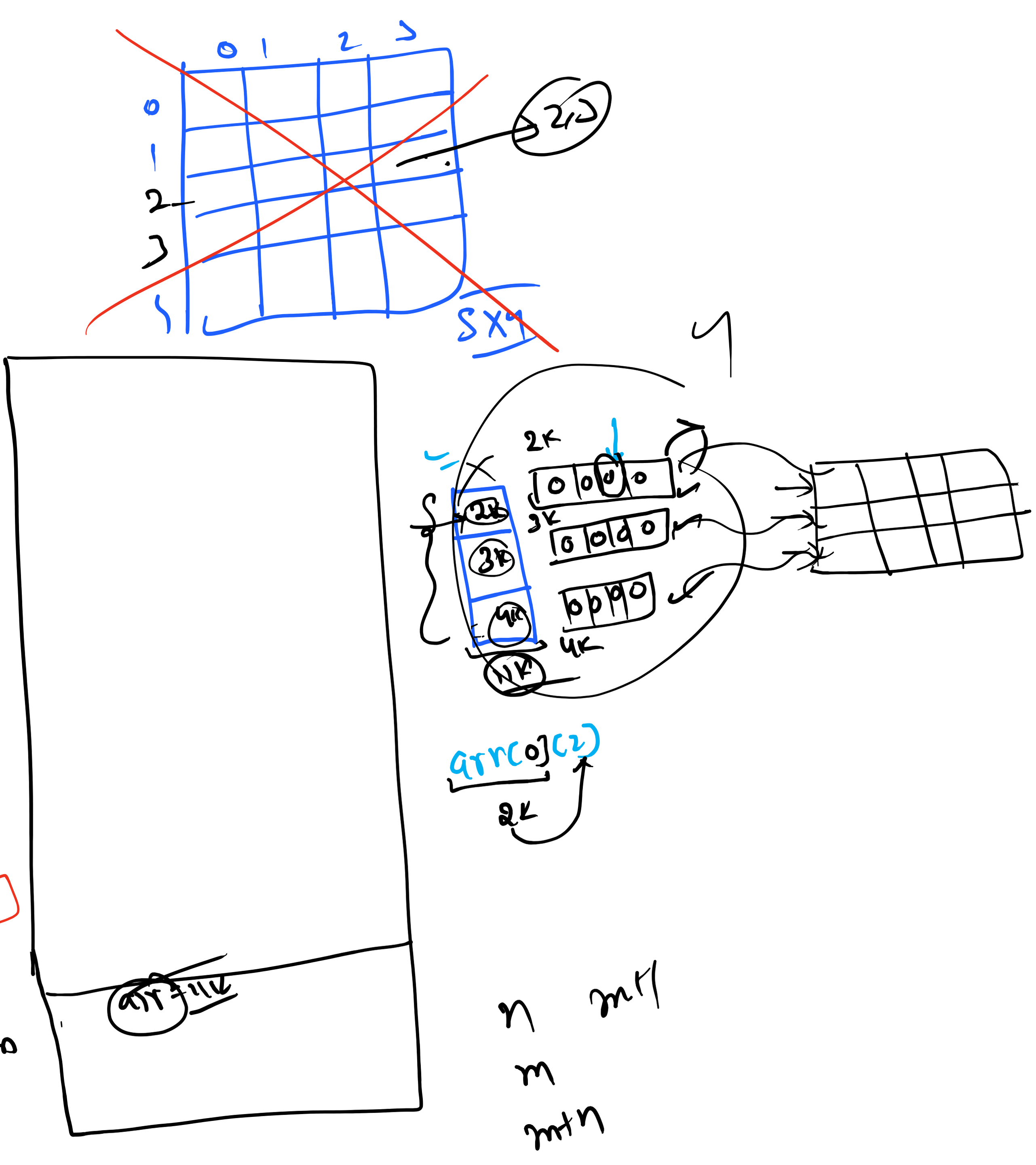
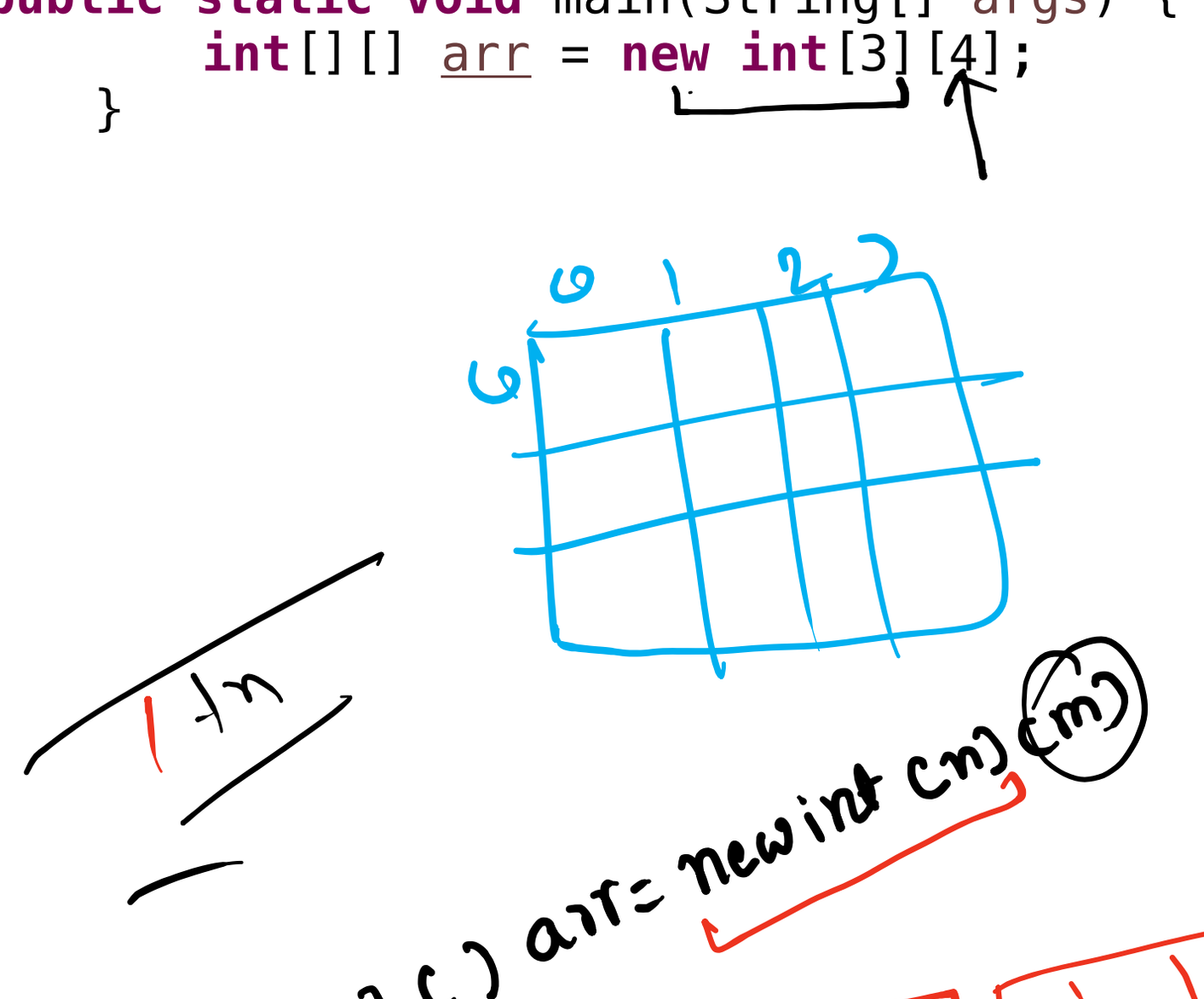
Lec-12

24/06/25 12:34 PM

int [][] arr

```java
public static void main(String[] args) {
        int[][] arr = new int[3][4];
    }
```

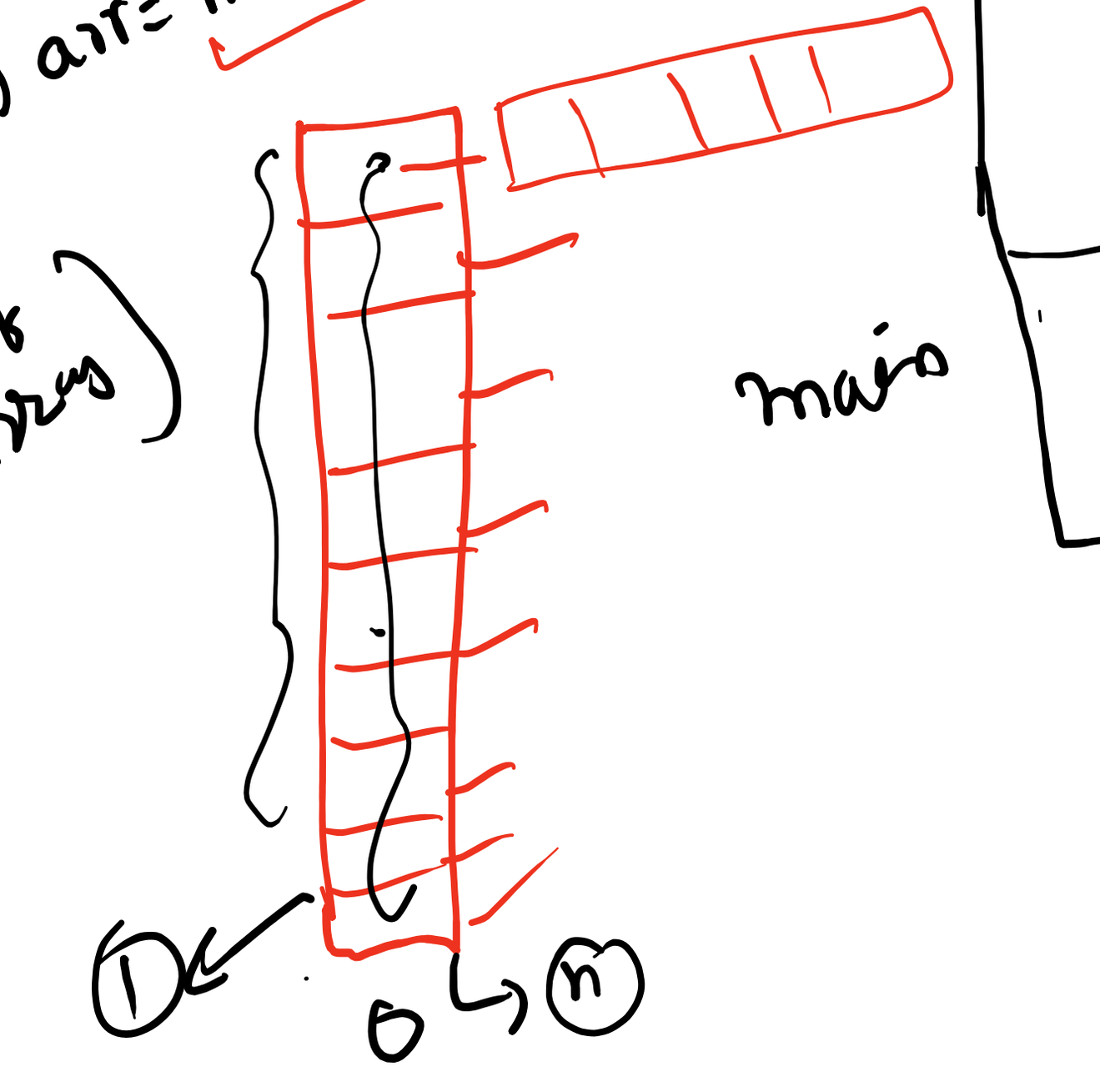




int [] [] arr = new int [n] [m]

Total no of 1 D array

main



n m+1

m

m+n

arr → Row → arr.length

→ col → arr[0].length

for(col=0 col<arr[0].length col++){

if(col%2==0){

row—

arr[col][row]

}

else

row+1—

}

0	1	2	3
1	2	3	4
5	6	7	8
15	16	17	18

1 5 15 16 6 2 3 7 17 18 8 4

(0-0) (0-1) 0-2 0-3

1-0 1-1 1-2 1-3

2-0 2-1 2-2 2-3

for(i=0 i<arr.len i++){

for(j=0 j<arr[0].len j++){

arr[i][j] = sc.nextInt()

Syso (arr[i][j])

}

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

(i,j) Row, col → col, Row

[0-1] → [1-0]

0-2 → [2-0]

0-3 → [3-0]

[1-2] → [2-1]

1-3 → (3-1)

(2-3) ⟷ (3-2)

for(i=0 i<arr.len i++){

for(j=i+1 j<arr[0].len

(i-j) — (j-i)

}

11<13

15>13

1	4	7	11	15
2	5	8	12	19
3	6	9	16	22
10	13	14	17	24
18	21	23	26	30

Stair-case Search

while (row<n && col>=0) {

if(arr[Row][col] == item){

return True

}

else if (arr[row][col] > item){

col--

}

else{

row ++

}

return False

1	4	7	11	15
2	5	8	12	19
3	6	9	16	22
10	13	14	17	24
18	21	23	26	30

1 4 7 11 15 19 22 24 30 26 23 21 18 10 3 2

5 8 12 16 17 14 13 6 9

for(i=minc i<=maxc i++){

Syso (arr[minR][i])

}

minR++

for(i=minR i<=maxR i++){

Syso (arr[i][maxc])

}

maxc--

for(int i=maxc i>=minc i--){

Syso (arr[maxR][i])

}

maxR--

for(i=maxR i>=minR i--){

Syso (arr[i][minc])

}

minc++

```java
public static void Print(int[][] arr) {
        int minr = 0, minc = 0, maxr = arr.length - 1, maxc = arr[0].length - 1;
        int te = arr.length * arr[0].length;
        int c = 0;
        while (c < te) {
            for (int i = minc; i <= maxc; i++) {
                System.out.print(arr[minr][i] + " ");
                c++;
            }
            minr++;
            for (int i = minr; i <= maxr; i++) {
                System.out.print(arr[i][maxc] + " ");
                c++;
            }
            maxc--;
            for (int i = maxc; i >= minc; i--) {
                System.out.print(arr[maxr][i] + " ");
                c++;
            }
            maxr--;
            for (int i = maxr; i >= minr; i--) {
                System.out.print(arr[i][minc] + " ");
                c++;
            }
            minc++;
        }
    }
```

minR = 0 1 2 maxR = 2 1 0

minc = 0 1 maxc = 3 2 1

0	1	2	3
1	2	3	4
5	6	7	8
9	10	11	12

c = 4+2+3+1 = 10+2 = 12

1 2 3 4 8 12 11 10 9 5 6 7 6

4567012

Input: nums = [4,5,6,7,0,1,2], target = 3

Output: -1

while (lo <= hi) {

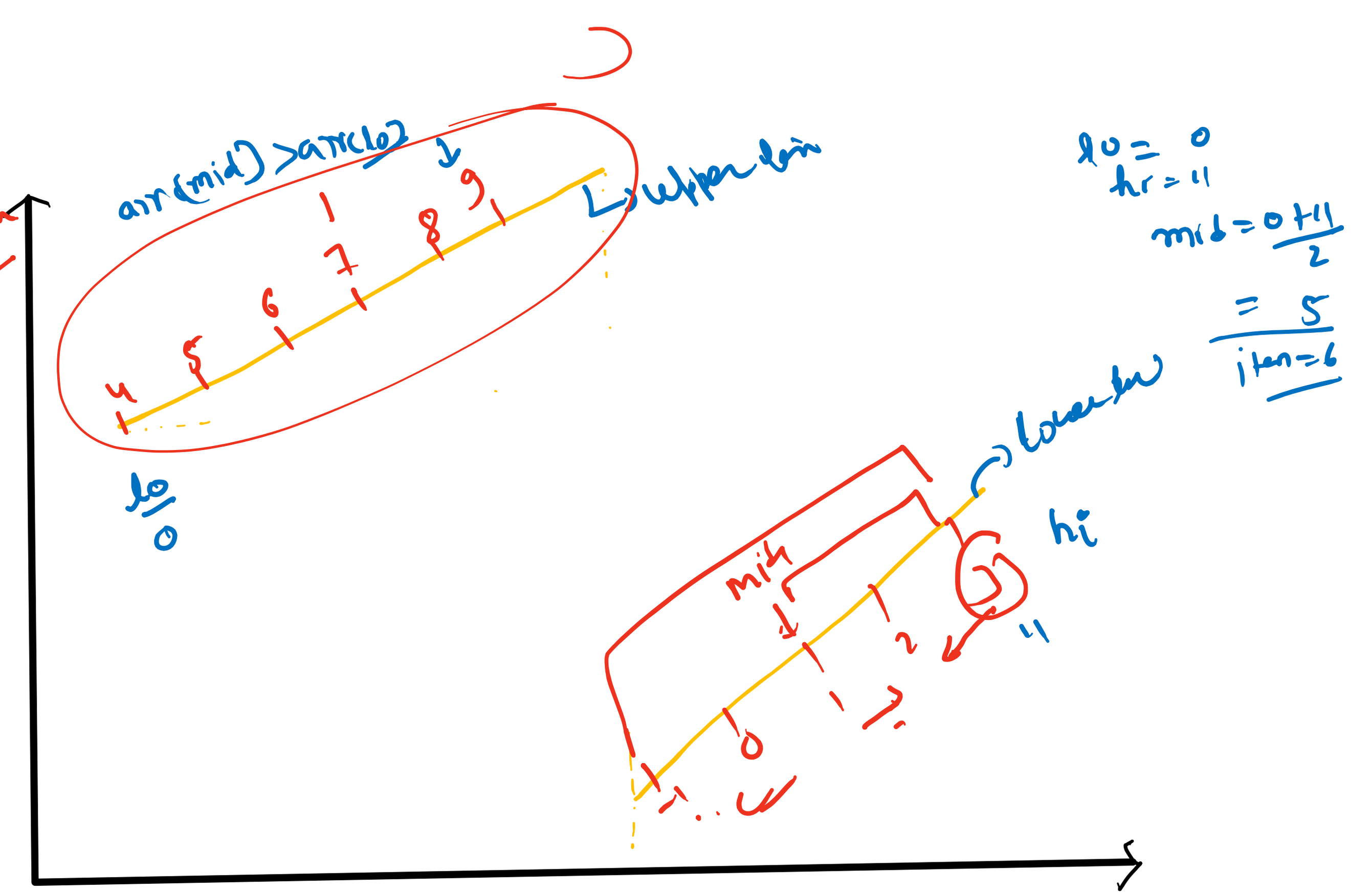
mid

int lo = 0

int hi = n-1

[4 5 6 7 8 9 -1 0 1 2 3]

item >= arr[lo] && arr[mid] > item



lo = 0

hi = 11

mid = (0+11)/2 = 5

item = 6