Agenda

1) Revise
2) Search in sorted rotated array
3) Given N, find sqrt(N)
4) Every element occurs twice except 1. Find it
5) Special integer

Q Sorted rotated array (Distinct ele)

rotated → End to front

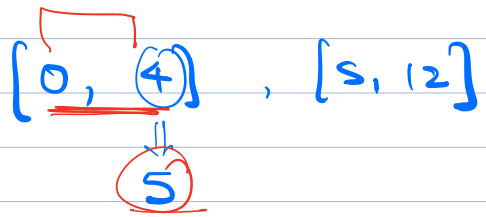
Search for a given element X.

A: -20, -14, -8, -4, 1, 2, 4, 7, 11, 14, 19, 23, 27

SRA: 11, 14, 19, 23, 27, -20, -14, -8, -4, 1, 2, 4, 7

0	1	2	3	4	5	6	7	8	9	10	11	12
11	14	19	23	27	-20	-14	-8	-4	1	2	4	7

s	e	mid	
0	12	6	A[6] < A[0] Discard right
0	5	2	A[2] > A[0] Discard left
3	5	4	



Code

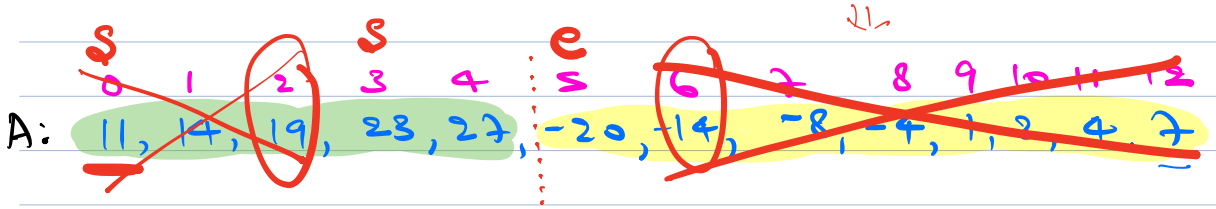
```
int search ( A[] , N , X) {
    if (A[0] < A[N-1]){
        // Apply 1 BS on the entire array
    }

    s = 0, e = N-1, mid = 0

    while (s <= e) {
        mid = (s+e)/2

        → if ( A[mid] > A[mid+1] &&
               A[mid] > A[mid-1]){
            // call 1 BS on A[0, mid]
            // call 1 BS on A[mid+1, N-1]
            return;
        }
        → if ( A[mid] < A[mid+1] &&
               A[mid] < A[mid-1]) {
            // call 1 BS on A[0, mid-1]
```

```
        // call 1 BS on A[mid, N-1]
        return;
    }
    if ( A[mid] < A[0] ) {
        // Discard right
        e = mid - 1;
    }
    else {
        // Discard left
        s = mid + 1;
    }
}
```



search (A, 13, -4)

s	e	mid	
0	12	6	A[6] < A[0] ▷R
0	5	2	A[2] > A[0] ▷L
3	5	4	

T.C. = O (log N)

Q Given a no. N. find out sqrt(N)??

N = 39 $\sqrt{39} = 6$

$7^2 = 49$ $6.1 = 6^2 = 36$

39

i = 1	1x1 = 1	< 39
i = 2	2x2 = 4	< 39
i = 3	3x3 = 9	< 39
i = 4	4x4 = 16	< 39
i = 5	5x5 = 25	< 39
i = 6	6x6 = 36	< 39 Ans
i = 7	7x7 = 49	> 39 X

Ans

break

sqrt(N) +1

i = 1 ans = 1

T.C. = $O(\sqrt{N})$

```
while ( i*i <= N ) {
    ans = i
    i++;
}
```

N → [1, N]

Answer space

BS on answer space

s = 1

e = N (39)

s e mid

1	39	20	20×20 = 400 > 39 Discard right
1	19	10	10×10 = 100 > 39 Discard right
1	9	5	5×5 = 25 < 39 Discard left ans = 5 Better ans in right
6	9	7	7×7 = 49 > 39 Discard right
6	6	6	6×6 = 36 < 39 ans = max(ans, 6) ans = 6 > L
7	6	✗	broke

T.C. $O(\log(N))$

Code

```
int sqrt(N) {
```

```
s = 1,   e = N;   mid
ans = 1;

while (s <= e) {
        mid = (s+e)/2

        if (mid x mid == N) {
                return mid;
        }
        if (mid x mid > N) {
                e = mid - 1;
        }
        else {
                ans = mid
                s = mid + 1;
        }
}
return ans;

}
```

Class starts at 11:00

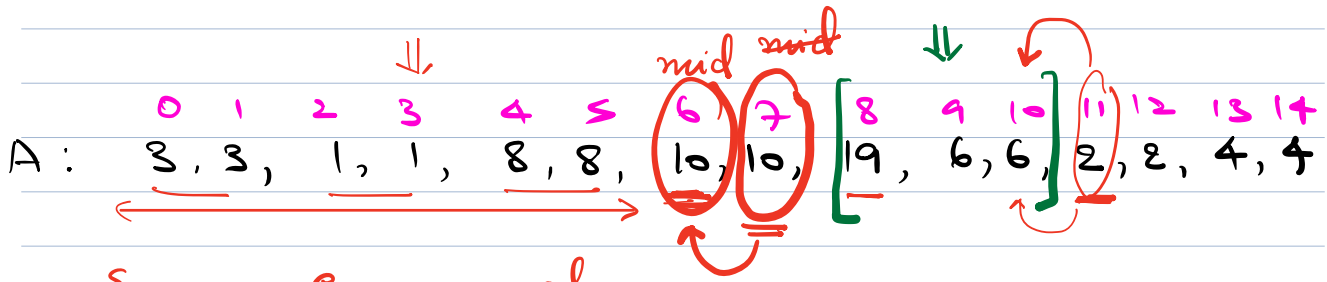
Q) Every element in an array occurs twice except 1.
Find the element.

XOR sol^{n} $\Rightarrow$ O(N)

NOTE: Duplicates are present adjacent to

each other.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A:	3,	3,	1,	1,	8,	8,	10,	10,	19,	6,	6,	2,	2,	4,	4



s	e	mid		
0	14	~~7~~ 6	×	A[s, mid-1] is even lengthed
+2				> L
8	14	11		A[s, mid-1]
		-1		Odd
				> R
8	10	9		A[8, 8] → 1
				↑ Odd
				> R
8	8	8	✓	

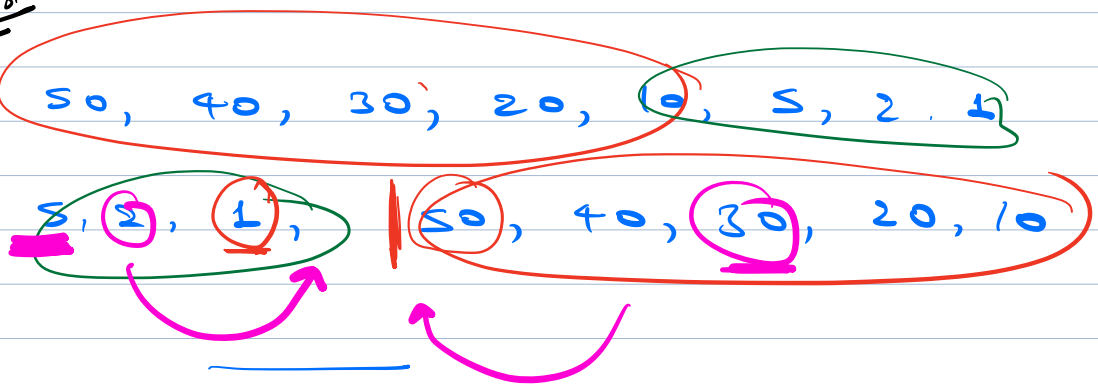
Code

```
int searchUnique (A[], N) {
    if (N==1) { return A[0]; }
    if (A[0] != A[1]) { return A[0]; }
    if (A[N-1] != A[N-2]) { return A[N-1]; }
```

```
s = 0,   e = N-1,   mid = 0
while ( s <= e) {
    mid = (s+e)/2
    if (A[mid] == A[mid-1]) {
        mid = mid-1;
    }
    if (A[mid] != A[mid+1]) {
        return A[mid];
    }
    // cal size of subarray from
    // s to mid-1
    size = (mid-1) - s + 1 = mid-s
    if (size % 2 == 0) {
        // DL
        s = mid+2;
    }
    else {
        e = mid-1;
    }
}
}
```

T.C. = $O(\log_2 N)$

Doubt



mid