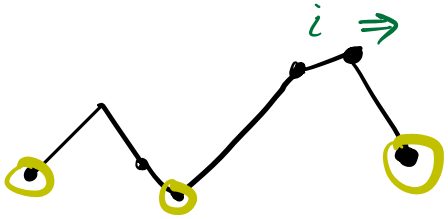
Q) Given an integer array of N distinct elements.
find any local minima.

A: [3, 6, 1, 0, 9, 15, 8]
(0, 1, 2, 3, 4, 5, 6)



A[i] < A[i-1]
A[i] < A[i+1]
> Local minima

A: [21, 20, 19, 17, 15, 9, 7]

A: [1, 2, 3, 4, 5]

A: [5, 4, 3, 2, 1]

Sol^n 1) Brute force

∀ i in [0, N-1]
check

if ((i==0 || A[i]<A[i-1]) && (i==N-1 || A[i]<A[i+1]))

T.C. = O(N)

↓ Optimise ??

2) Binary Solution

A[mid] < A[mid-1] && < A[mid+1] ⇐ → Mid is a minima

↳ Go to left

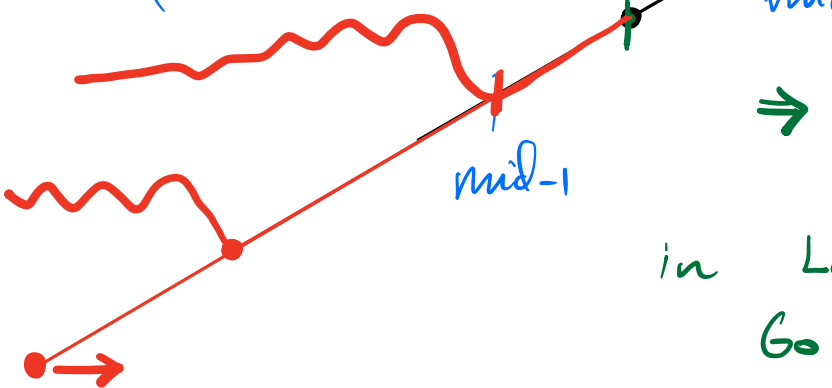
↳ Go to right

Obs: All elements are distinct

Case I ⇒ Mid is Minima

⇒ Mid

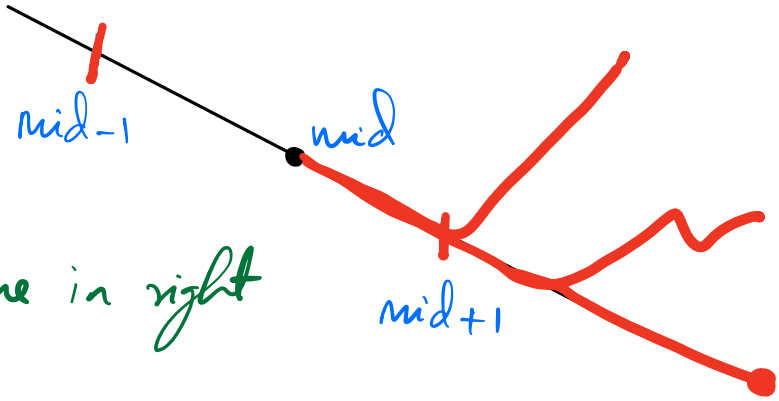
Case II (A[mid-1] < A[mid] mid mid+1 [A[mid+1] > A[mid])



⇒ Atleast one minima is present in Left

Go to Left

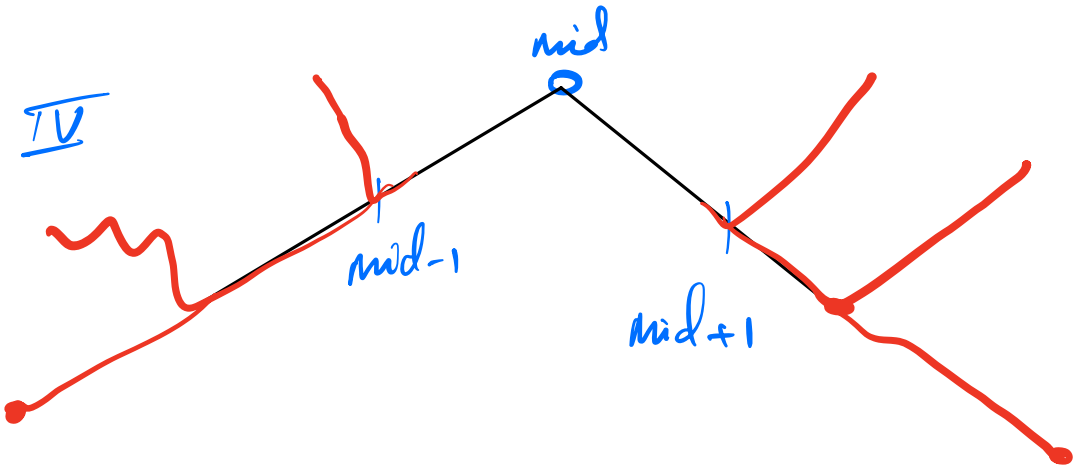
Case III



Atleast one minima in right

Go to right

Case IV



Code

```
int findMinima(A, N){

    S=0;
    e= N-1;

    while (s<=e){

        mid = s+ (e-s)/2;
        if ((mid==0 || A[mid-1]>A[mid]) &&
            (mid==N-1 || A[mid+1]> A[mid])){
```

```
            return mid;
        }
        else if (A[mid-1] < A[mid]) {
            // Go to left
            e = mid-1;
        } else {
            // Go to right
            s = mid+1;
        }
    }
}
```

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

S.C. = $O(1)$

BS

```
s = 0
e = N-1;        } Define initial Search Space

while (s<=e) {
```

Case I ⇒ When mid is the ans.

Case II ⇒ When to go left

Case III When to go right

}

Search in Sorted Rotated Array

$$A = [\underset{0}{4}, \underset{1}{5}, \underset{2}{6}, \underset{3}{7}, \underset{4}{8}, \underset{5}{9}, \underset{6}{1}, \underset{7}{2}, \underset{8}{3}]$$

Q) Given a sorted rotated array. Find a given element K.

$$A = [\underset{0}{4}, \underset{1}{5}, \underset{2}{6}, \underset{3}{7}, \underset{4}{8}, \underset{5}{9}, \underset{6}{1}, \underset{7}{2}, \underset{8}{3}]$$

$K = 7 \Rightarrow 3$

Soln 1) Brute force

$\Rightarrow$ Linear Search

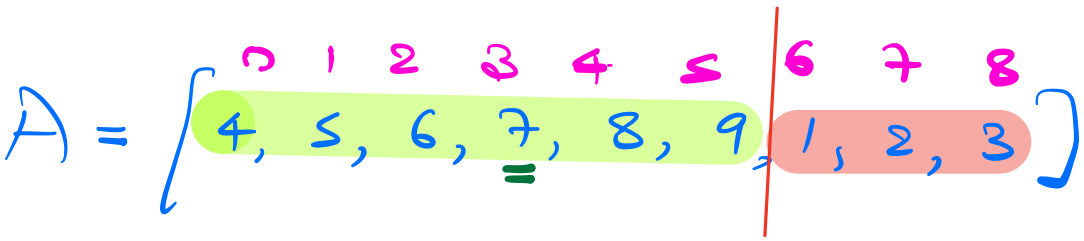
T.C. = $O(N)$

2) Binary Search

1) When mid is ans $\Rightarrow$ (A[mid] == K)

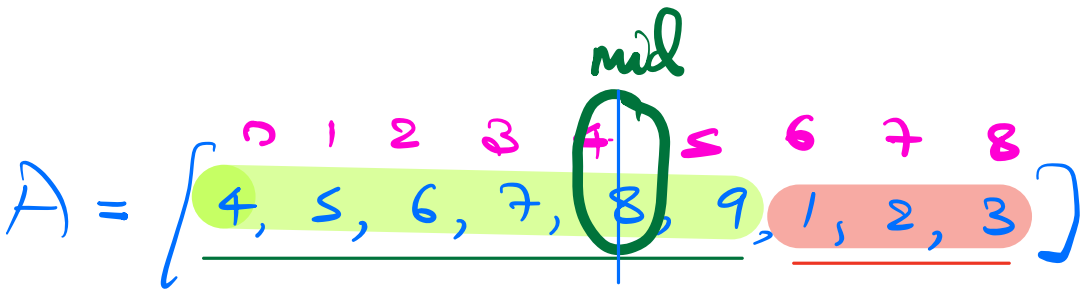
2) When to go left

2) When to go right

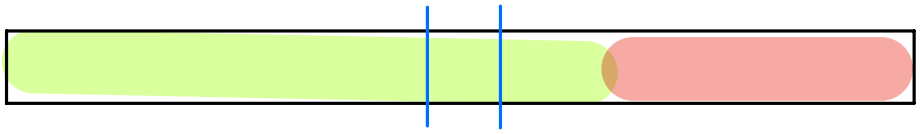


H.W. find where the array is rotated.

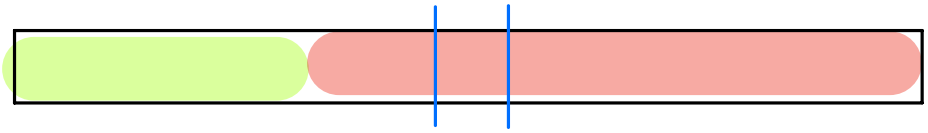
$\Rightarrow \log_2 N$



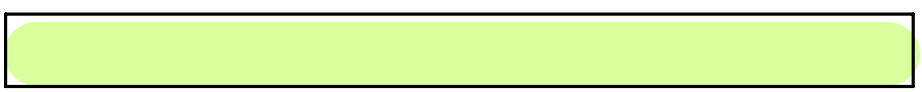
I



II



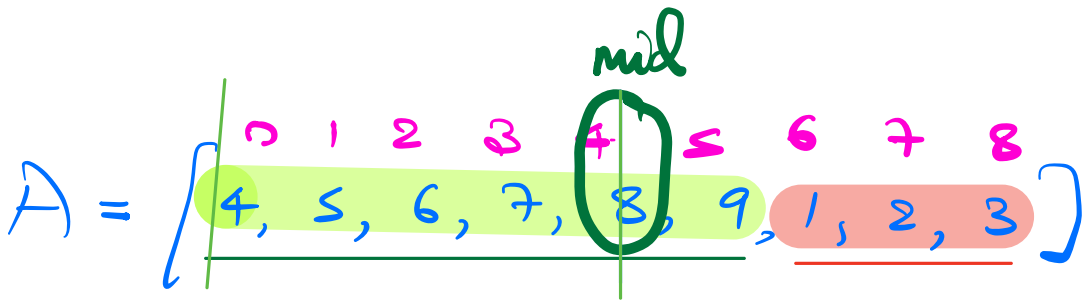
III

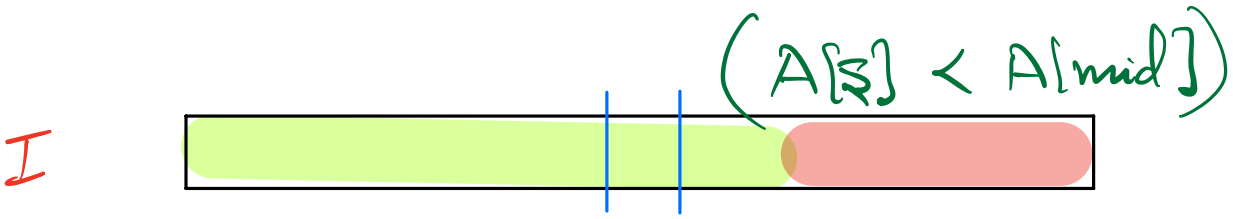


A[0] < A[N-1] rotated 0 or $c \times N$ times

sorted

$\Rightarrow$ Directly apply BS.



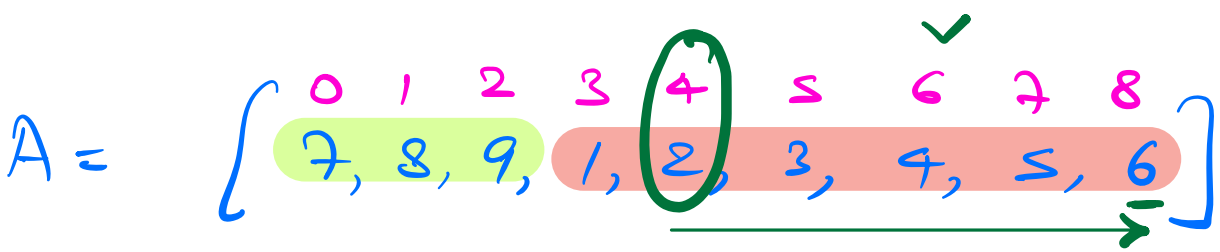


$K = 6$ $A[mid] = 8$ $K = 2$

Obs 1 s to mid are increasing

$K \Rightarrow [A[s], A[mid]] \Rightarrow$ Go to Left

Else $\Rightarrow$ Go to right



$A[e] > A[mid]$

$K \Rightarrow [A[mid], A[e]] \Rightarrow$ Go to right

Else $\Rightarrow$ Go to Left.

Code

```
int searchInRotatedArray(A, N) {

    s = 0;
    e = N-1;

    while (s <= e) {
        mid = s + (e-s)/2;
        if (A[mid] == K) { return mid; }

        if ( A[s] <= A[mid] ) { // s to mid is sorted
            if ( K < A[mid] && K >= A[s]) {
                // Go to left
                e = mid - 1;
            }
            else { // Go to right
                s = mid + 1;
            }
        } else { // mid to e is sorted.
            if ( K > A[mid] && K <= A[e]) {
                // Go to right
                s = mid+1;
            } else { // Go to left
                e = mid - 1;
            }
        }
    }
```

}

return -1;

}

K = 7

A = [0 1 2 3 4 5 6 7 8
4, 5, 6, 7, 8, 9, 1, 2, 3]

s	e	mid	
0	8	4	⇒ A[s] < A[mid] ⇒ Sth mid is sorted.

K ≥ A[0] && K < A[mid]
⇓ Left

0	3	1	⇒ A[s] < A[mid] ⇒ Left is sorted

~~K > A[s] && K > A[mid]~~
Right

2	3	2

Given an integer N.
find the value of square root of N.

⇒ Binary Search on Answer Space

Eg $N = 36 \Rightarrow 6$

$N = 49 \Rightarrow 7$

$\vdots$

$N = 52 \Rightarrow 7$ $(7.\_\_)$

$N = 59 \Rightarrow 7$ $(7.\_\_)$

$\vdots$

$N = 63 \Rightarrow 7$ $(7.\_\_)$

$N = 64 \Rightarrow 8$

Sol^n) Brute force

Largest no. whose square $\leq N \Rightarrow$ Ans

$N = 63$

1	2	3	4	5	6	7	8
1	4	9	16	25	36	49	64

$(\sqrt{N}+1)$

```
ans = 1;
for (i = 1; i <= N; i++) {
    if (i*i <= N)
        ans = i;
    else
        break;
}
```

T.C. = O(sqrt(N))

⇓ Optimise

Observation

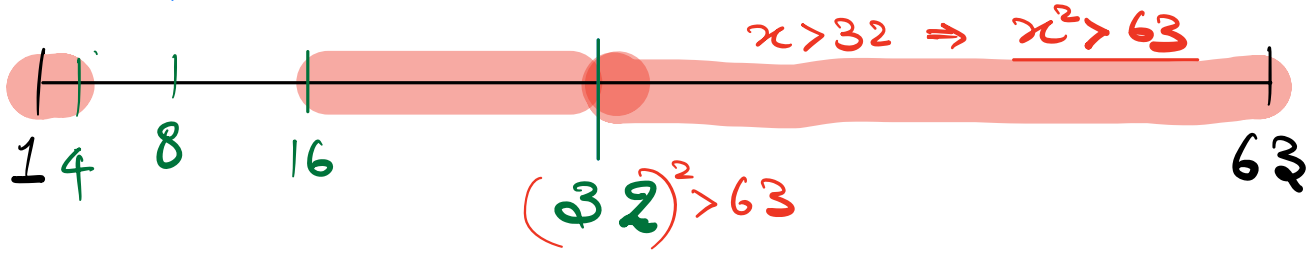
Answer Space. (continuous)

$N \xrightarrow{sqrt(N)} [1, N]$

⇑

⇓ sorted

$N = 63$



s	e	mid	
1	63	32	⇒ $(32)^2 > 63(N)$ Go to Left
1	31	16	⇒ $(16)^2 > 63(N)$ Go to left
1	15	8	⇒ $(8)^2 > 63$ Go to left
1	7	4	⇒ $(4)^2 < 63$ ⇓ $(4+1)^2 < 63$ ⇒ 4 is not ans Go to R
5	7	6	⇒ $(6)^2 < 63$ ⇓

$(6+1)^2 < 63 \Rightarrow$ 6 is not ans.

---

7 7 7 $\Rightarrow$ $(7)^2 < 63$

$\Downarrow$

$(7+1)^2 = 8^2 \not< 64 > 63$

Code

```
int sqrt(N) {

    s = 1
    e = N;

    while (s <= e) {

        mid = s + (e-s)/2;

        if (mid*mid <= N) {
            if ((mid+1) * (mid+1) > N) {
                return mid;
            } else {
                s = mid+1;
            }
        } else { // mid^2 > N => Go to left
```

```
        e = mid - 1;
    }
}
}
```

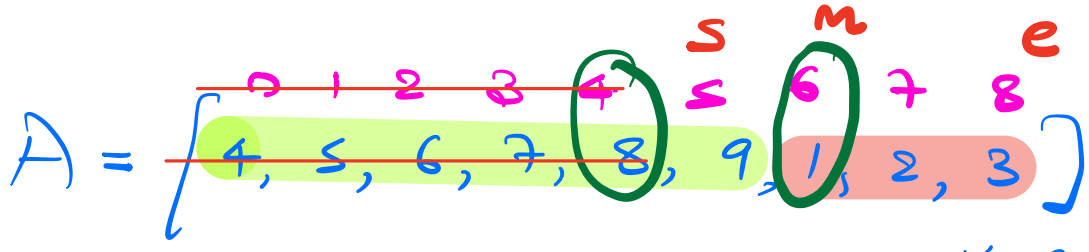
T.C. = $O(\log_2 N)$

S.C. = $O(1)$

$10^{10} \Rightarrow \sqrt{10^{10}} = 10^5$

$\log(10^9) \approx$ ~~40~~ 32

K = 9



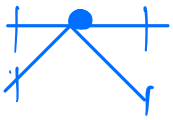
K = 9 / K = 2

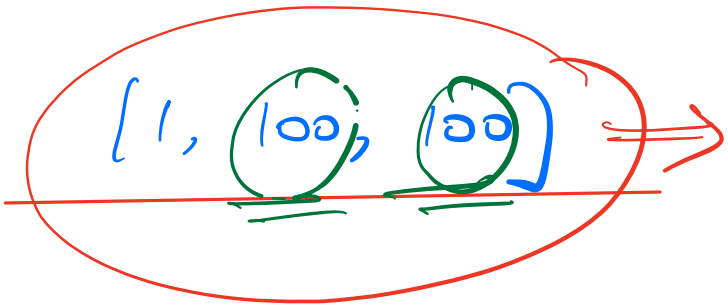
s	e	mid	
0	8	4	$\Rightarrow$ A[0] < A[4] → s to mid is Sorted. Right

5	8	6	if (A[s] < A[6]) ✗ s to mid is not Sorted. mid to e is Sorted. Right

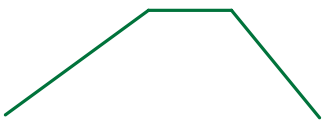
---

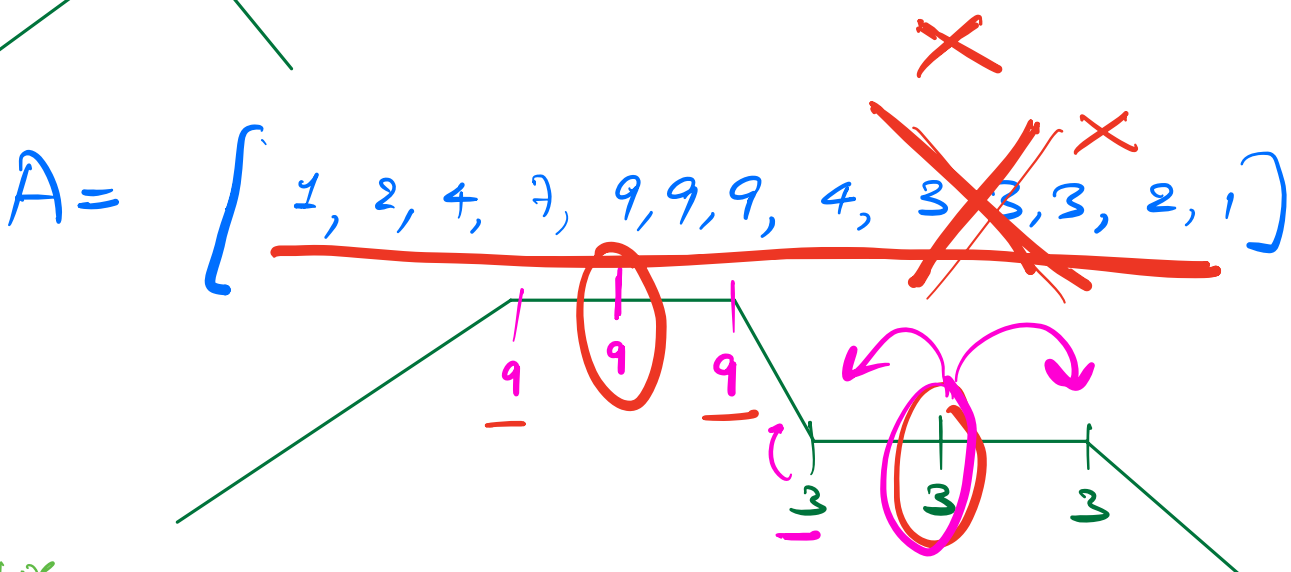
7	8	7	$\Rightarrow$ A[7] = 2





A: [1, 5, 7, 10, 10, 8, 5]





$\frac{0+x}{2} = 3$

$= 6$

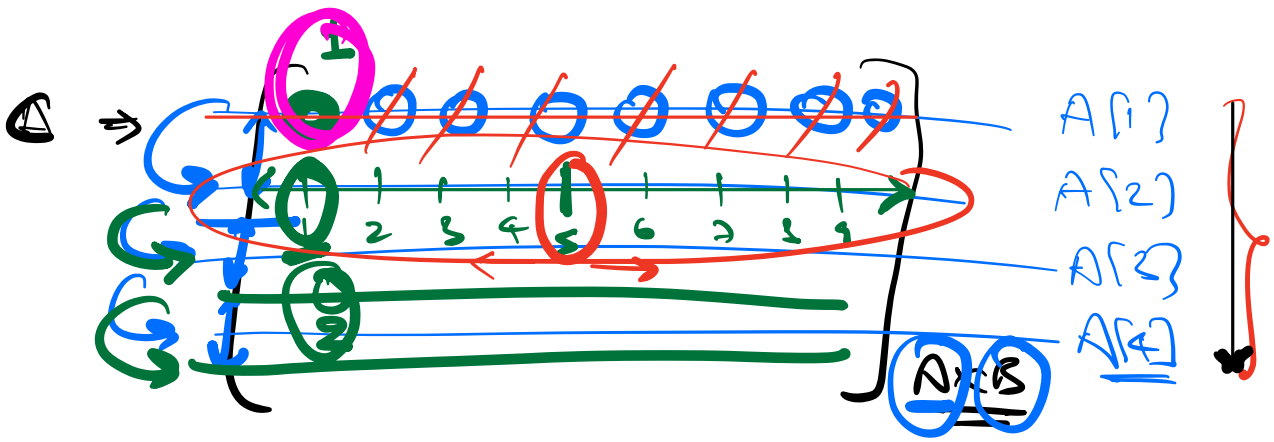
A= [1, 2, 3, 3, 3, 9, 5]

0 1 2 3 4 5 6

A= [5, 6, 2, 2, 2, 9, 5]

$A = [\, 5, 6, 9, 9, 9, 4, 3 \,]$

9



$\longleftrightarrow$

$B \log B + A \times (A-1)(\log B)$

$x - y$

mid ($\leq$) $B \log B + A^2 \log B$

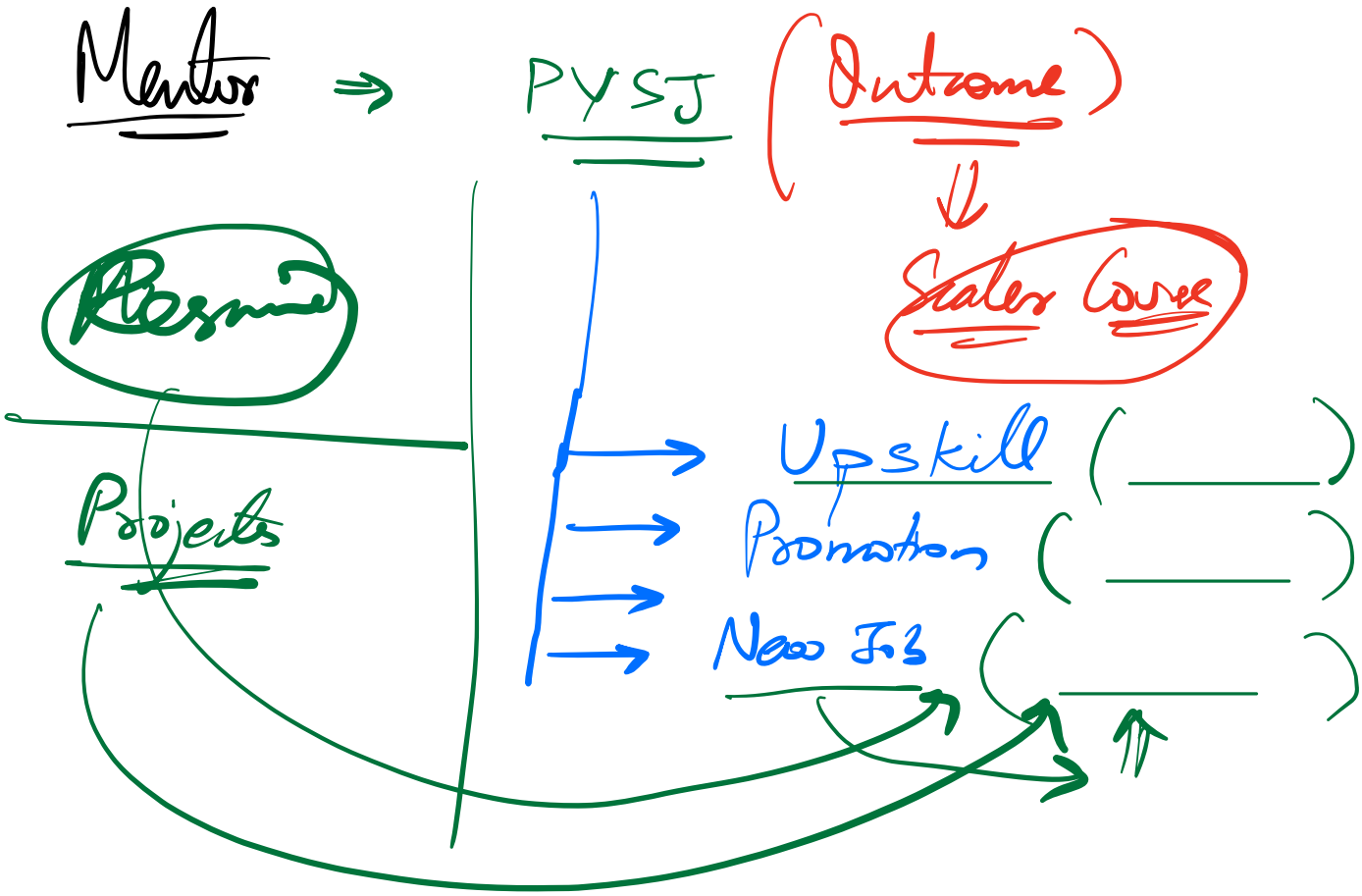
$(\leq 1) = 1$

$B \log B (A + A^2)$

$A^2 \log B$

$A = [\, 7, 3 \to 3 \;(\updownarrow 1),\ 2, 1 \to 2,\ 4, 9 \to 4 \;(\updownarrow 2) \,]$

1


Mentor
PYSJ
Outcome
Resume
Projects
Upskill
Promotion
New Job