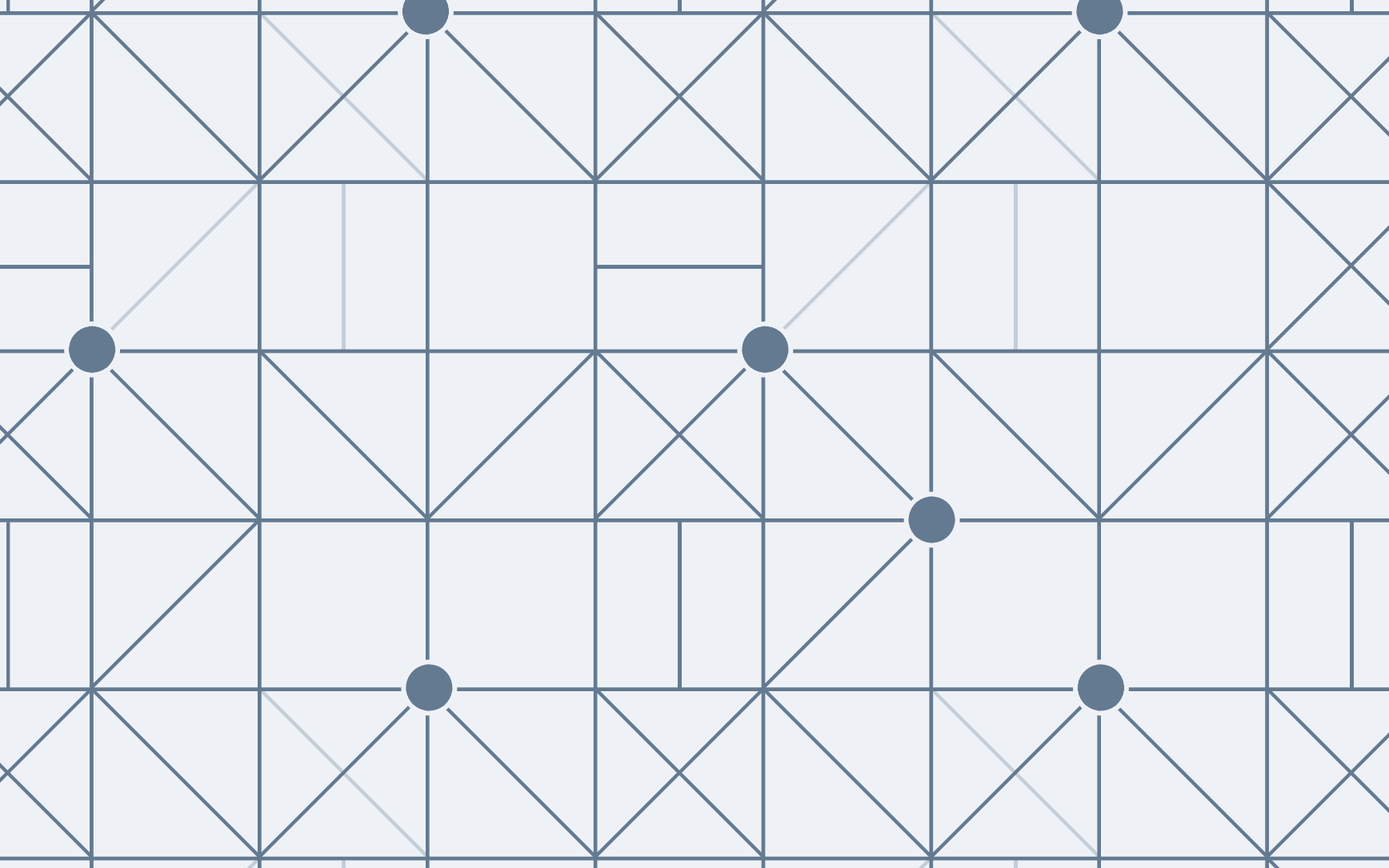


Binary Search

{ Searching Algorithm }

length region of search

✓TC : $O(\log_2 N)$ ✓SC : $O(1)$

- define range of search
- divide into half
- try to eliminate one & take another

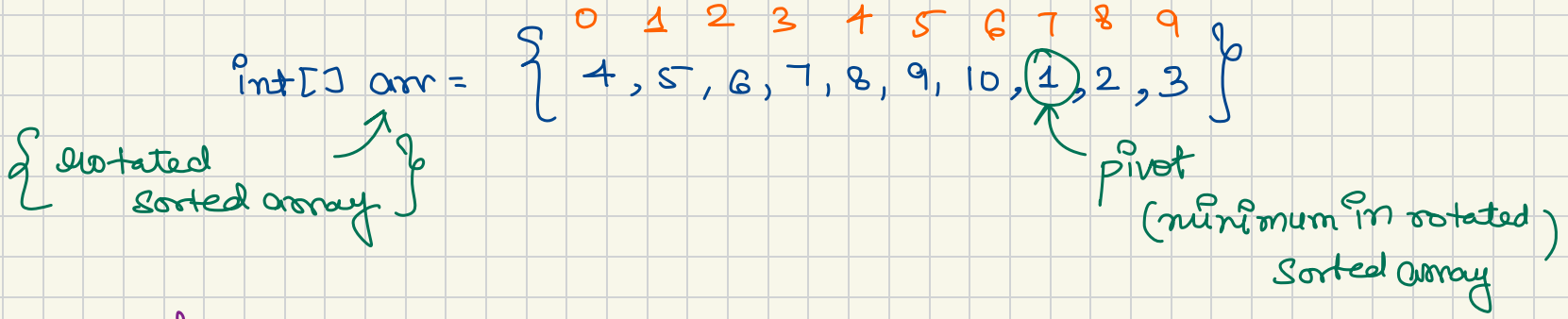
Agenda

- BS over ✓inc. array, or ✓non. dec. array.
- ✓BS over a Rotated Sorted array.
- ✓BS over a mountain array
- ✓BS over solution

↓

LC: Hard, Medium (Harder Side)

Search Pivot in a rotated Sorted Array.

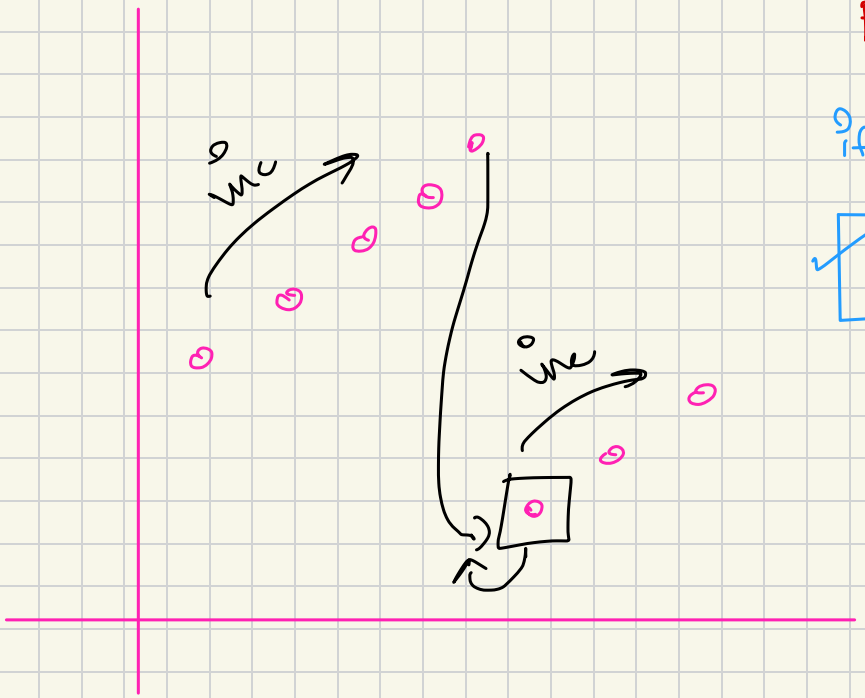


Brute force

Linear Search { find Min } TC : O(N) SC : O(1) ✓

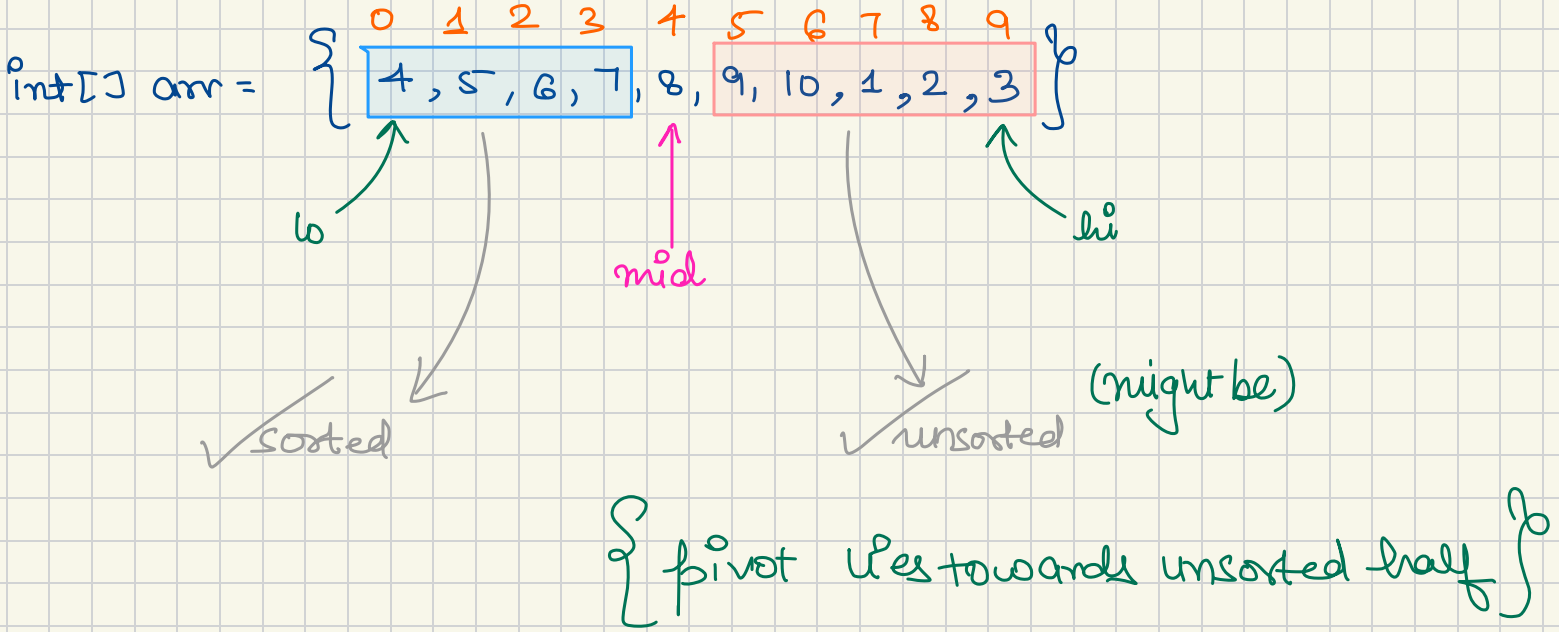
```
int[] arr = { 4, 5, 6, 7, 8, 9, 10, 1, 2, 3 }
            0  1  2  3  4  5  6  7  8  9
```

pivot



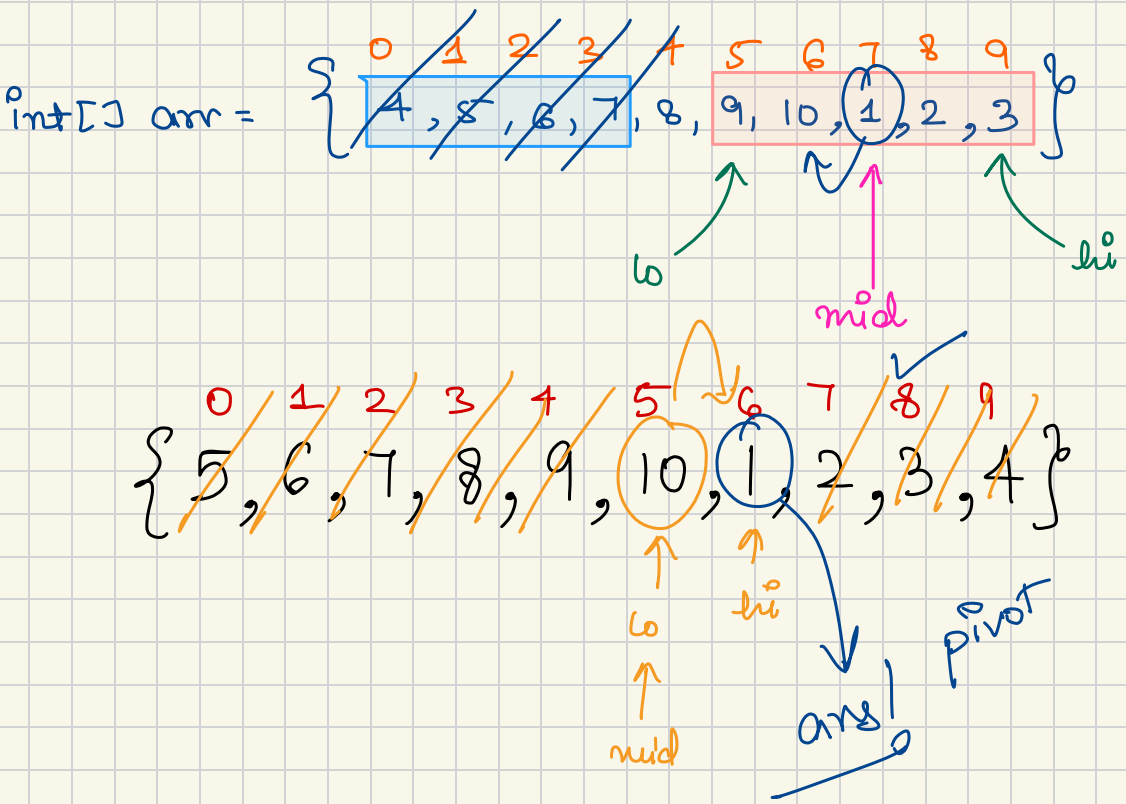
if pivot is a x index,

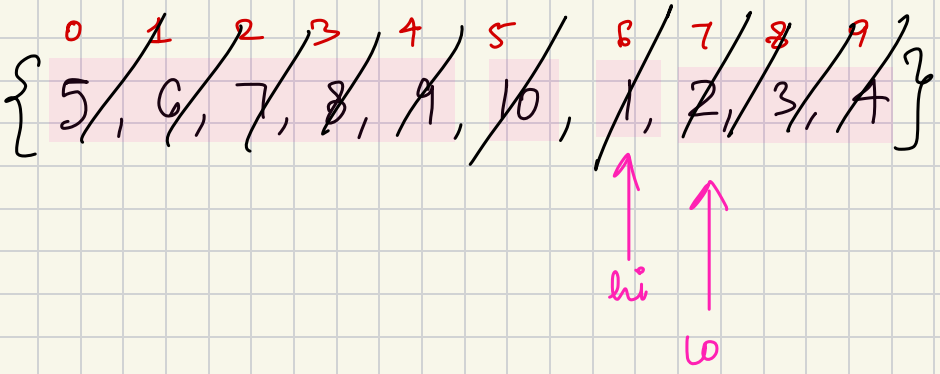
arr[x-1] > arr[x]



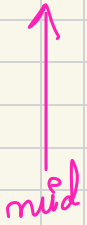
{ pivot lies towards unsorted half }

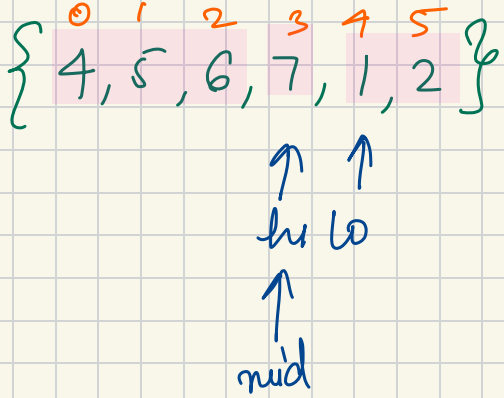
if (arr[s] < arr[e]) array is sorted in this scenario

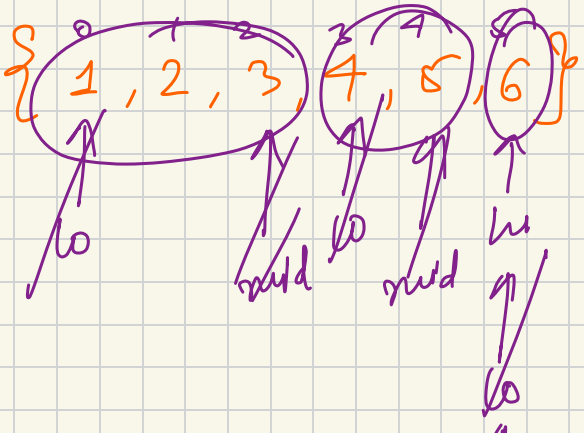

int[] arr =
0 1 2 3 4 5 6 7 8 9
4, 5, 6, 7, 8, 9, 10, 1, 2, 3
lo
mid
hi
0 1 2 3 4 5 6 7 8 9
5, 6, 7, 8, 9, 10, 1, 2, 3, 4
lo
mid
hi
pivot
ans!



pans = +~~0~~ ~~5~~ ~~2~~ 1 ✓







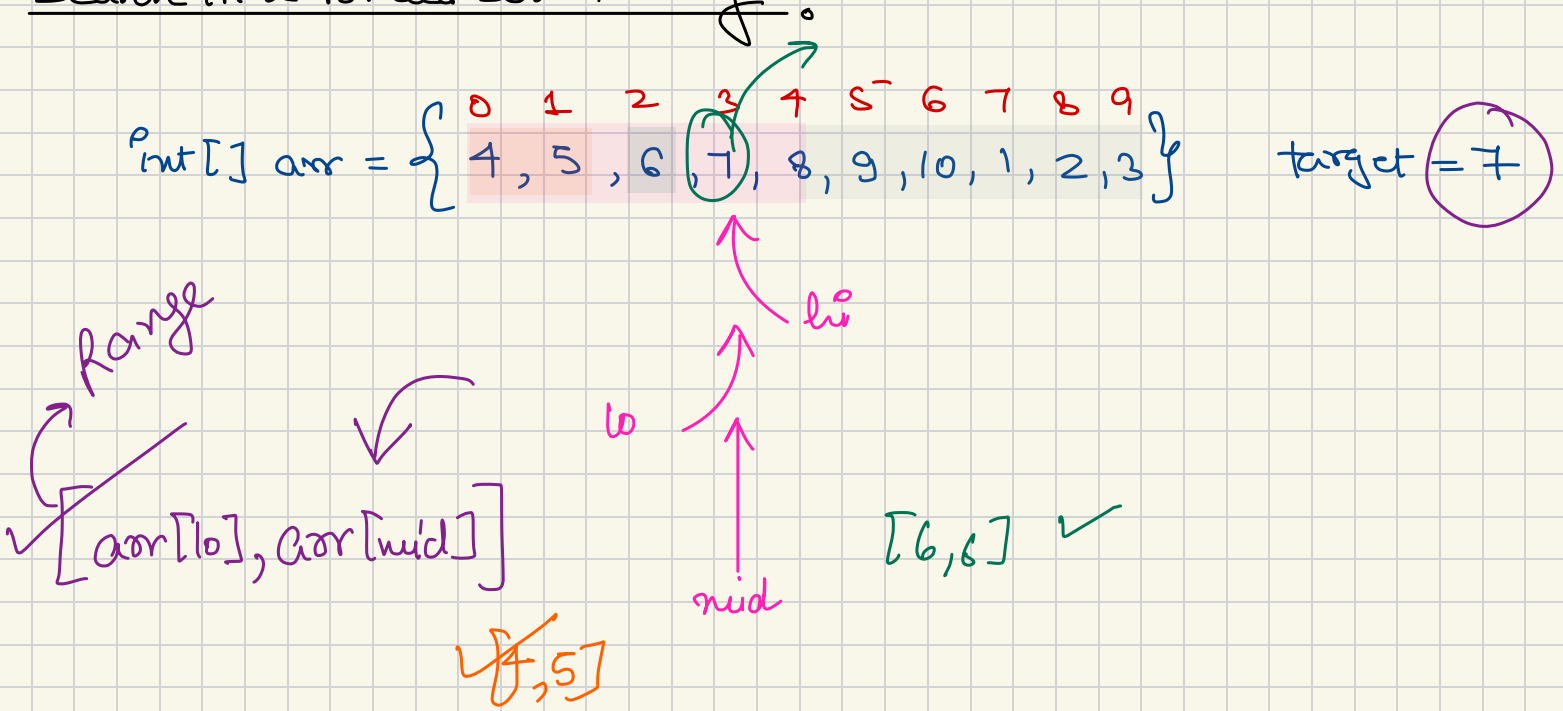
ans = +∞ ~~1~~ 1 ✓

{ ans = ~~+∞~~ 1 } mid

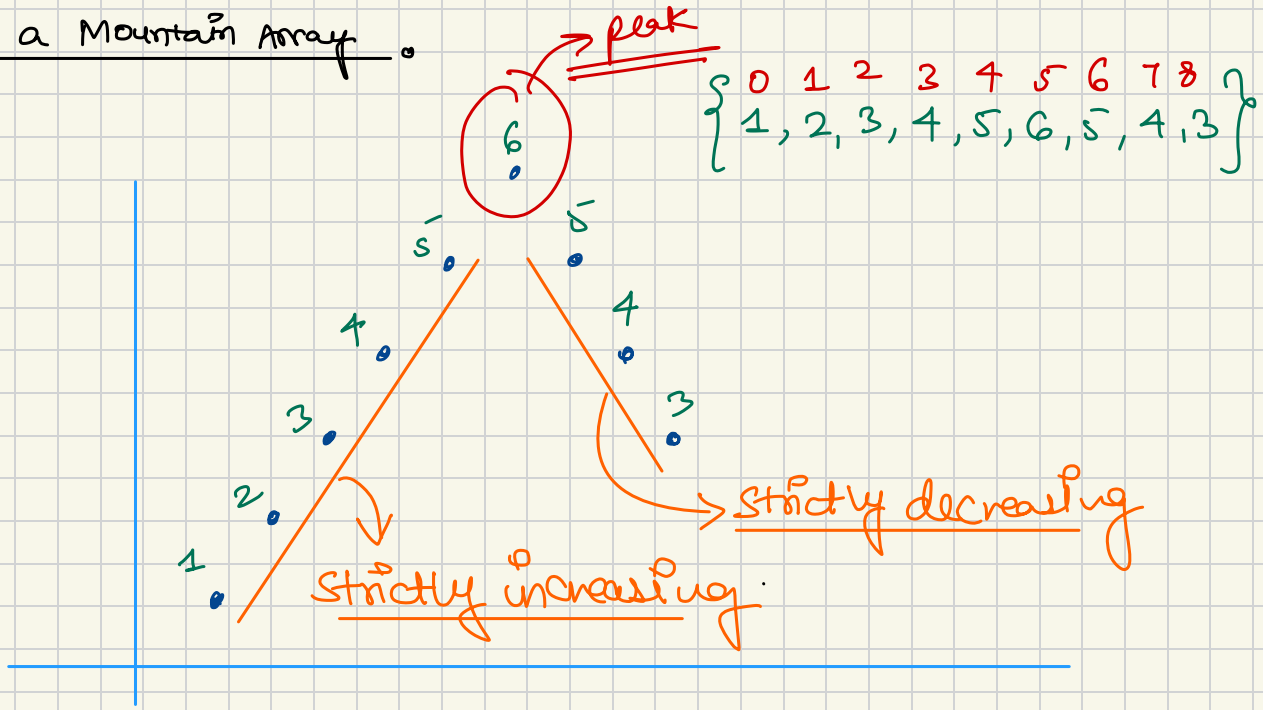
==

✓

Search in a rotated sorted array.

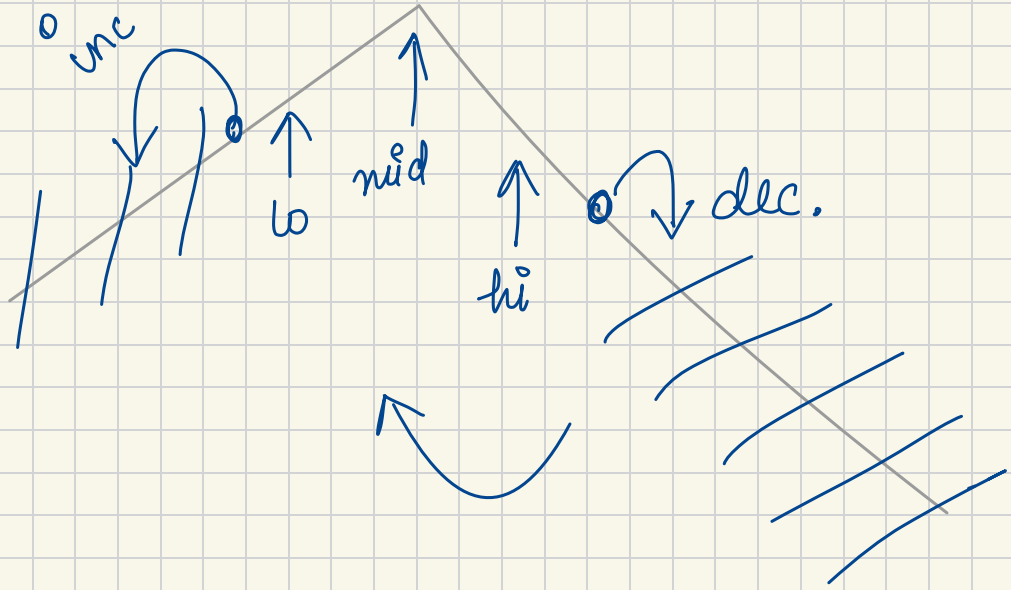


Peak in a Mountain Array



x^{th} index is peak

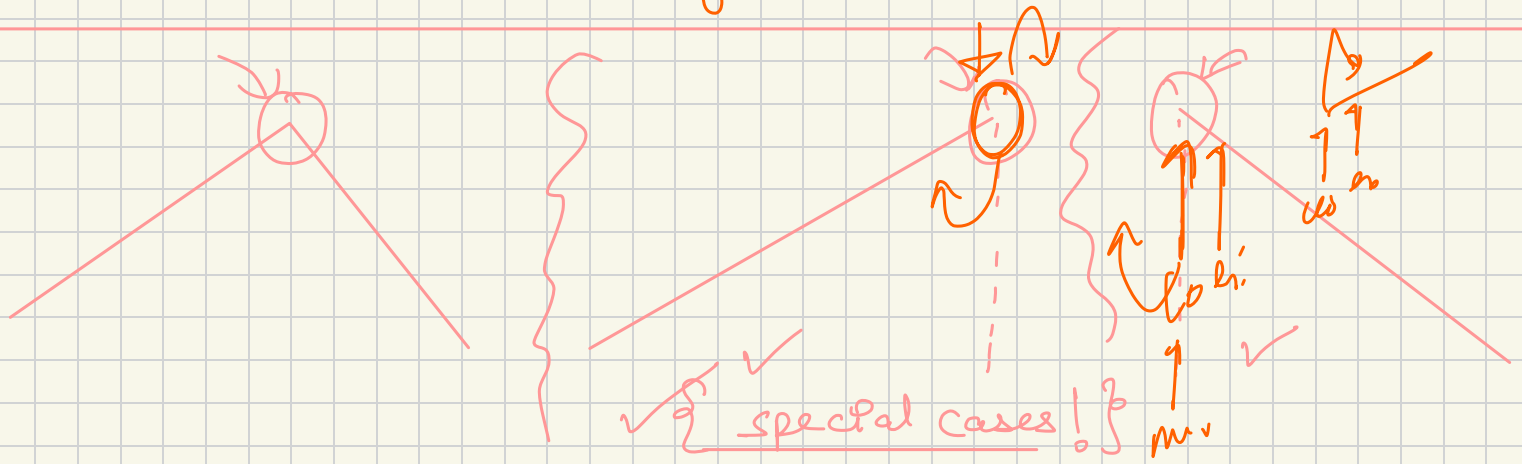
if (arr[x-1] < arr[x] && arr[x+1] < arr[x])


inc
lo
mid
hi
dec.

Case 1 : mid is peak

Case 2 : left side inc.
move right

Case 3 : right side is dec.
move left



Allocate minm Numbers of Pages

Books[] = { 0: 34, 1: 12, 2: 67, 3: 90 } Students = 2

- distribute these N books among M students
- such that each student gets min^m one book.
- book distribution should in contegeneous manner.

Books[] = { 0: 34, 1: 12, 2: 67, 3: 90 }

Students = 2

way 1

S1 → 34, 12, 67 → 113 pages

S2 → 90 → 90 pages

way 2

S1 → 34, 12 → 46 pages

S2 → 67, 90 → 157 pages

way 3

S1 → 34 → 34 pages

S2 → 12, 67, 90 → 169 pages

max^m Pages

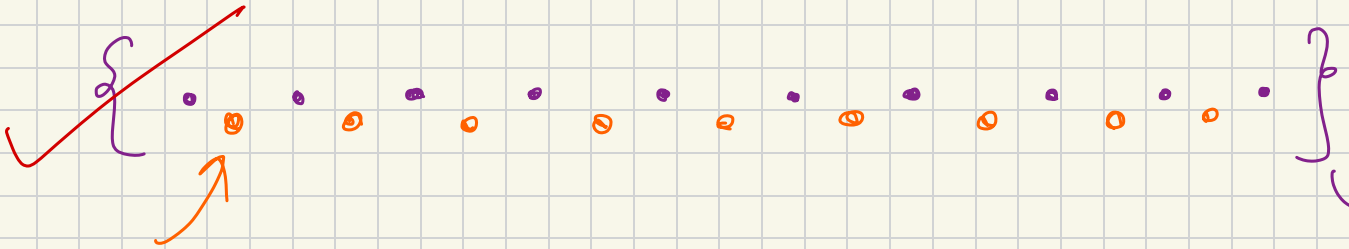
113 pages

157 pages

169 pages

113 pages

ans ✓



N Books.

M students → (M-1) seperators.

5 students → 5 clusters of these Books

dynamic programming

TC: O(N²) X

in (N-1) pos we have to place (M-1) seperators

TC: Exponential !

Books[] = { 0: 34, 1: 12, 2: 67, 3: 90 } Students = 2

Case 1 : if M = 1 {Student No.}

way 1 S1 → 34, 12, 67, 90 → 203 (min) → 203 pages.

Case 2 : if M = N

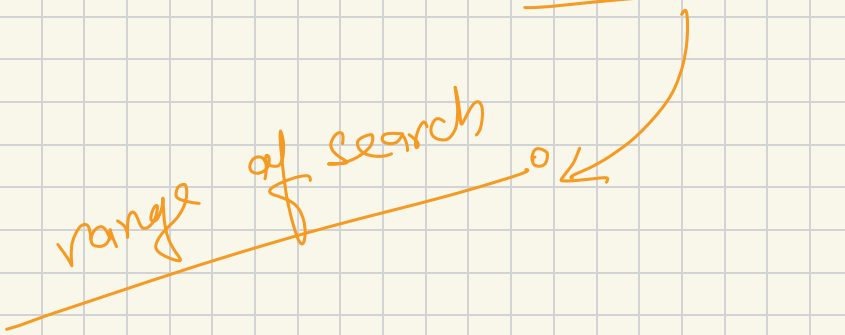
way 1 :

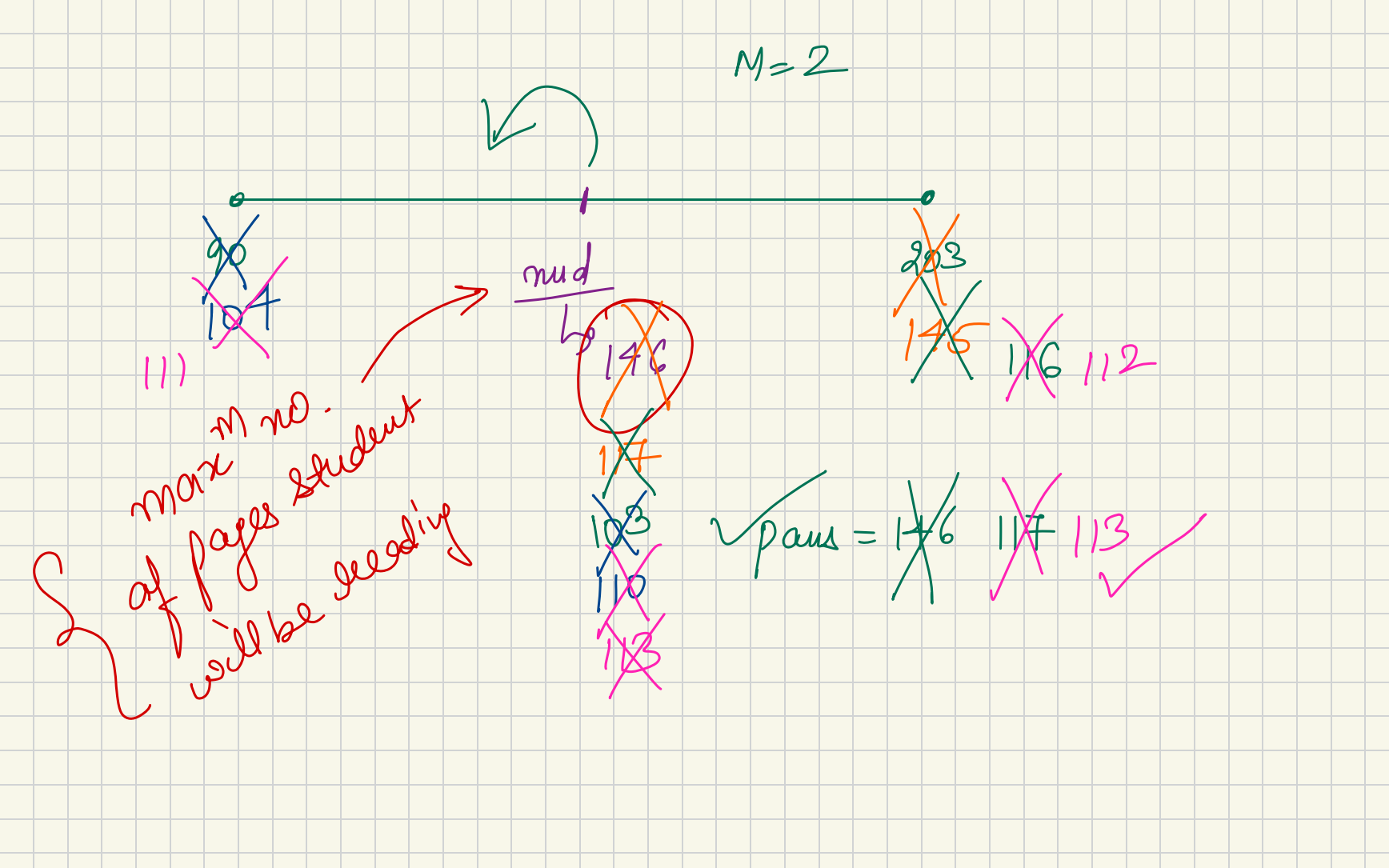
S1 → 34
S2 → 12
S3 → 67
S4 → 90

→ 90 (min) → 90 pages.

for $1 \leq M \leq N$

min of max pages = [90, 203]




M=2
90
104
111
mid
146
177
103
110
113
203
145
116
112
max no. of pages student will be reading
ans = 146 117 113

Books[] = { 0: 34, 1: 12, 2: 67, 3: 90 }

$max^m = 146$ ✓

S1 → 34 + 12 + 67

S2 → 90

Books[] = { 0: 34, 1: 12, 2: 67, 3: 90 }

$max^m = 117$ ✓

S1 → 34 + 12 + 67

S2 → 90

Books[] = { 0: 34, 1: 12, 2: 67, 3: 90 } $max^m = 103$

S1 → 34 + 12
S2 → 67
S3 → 90

Books[] = { 0: 34, 1: 12, 2: 67, 3: 90 } $max^m = 110$

S1 → 34 + 12
S2 → 67
S3 → 90

Books[] = { 0: 34, 1: 12, 2: 67, 3: 90 }

$max^m = 113$

S1 → 34 + 12 + 67

S2 → 90