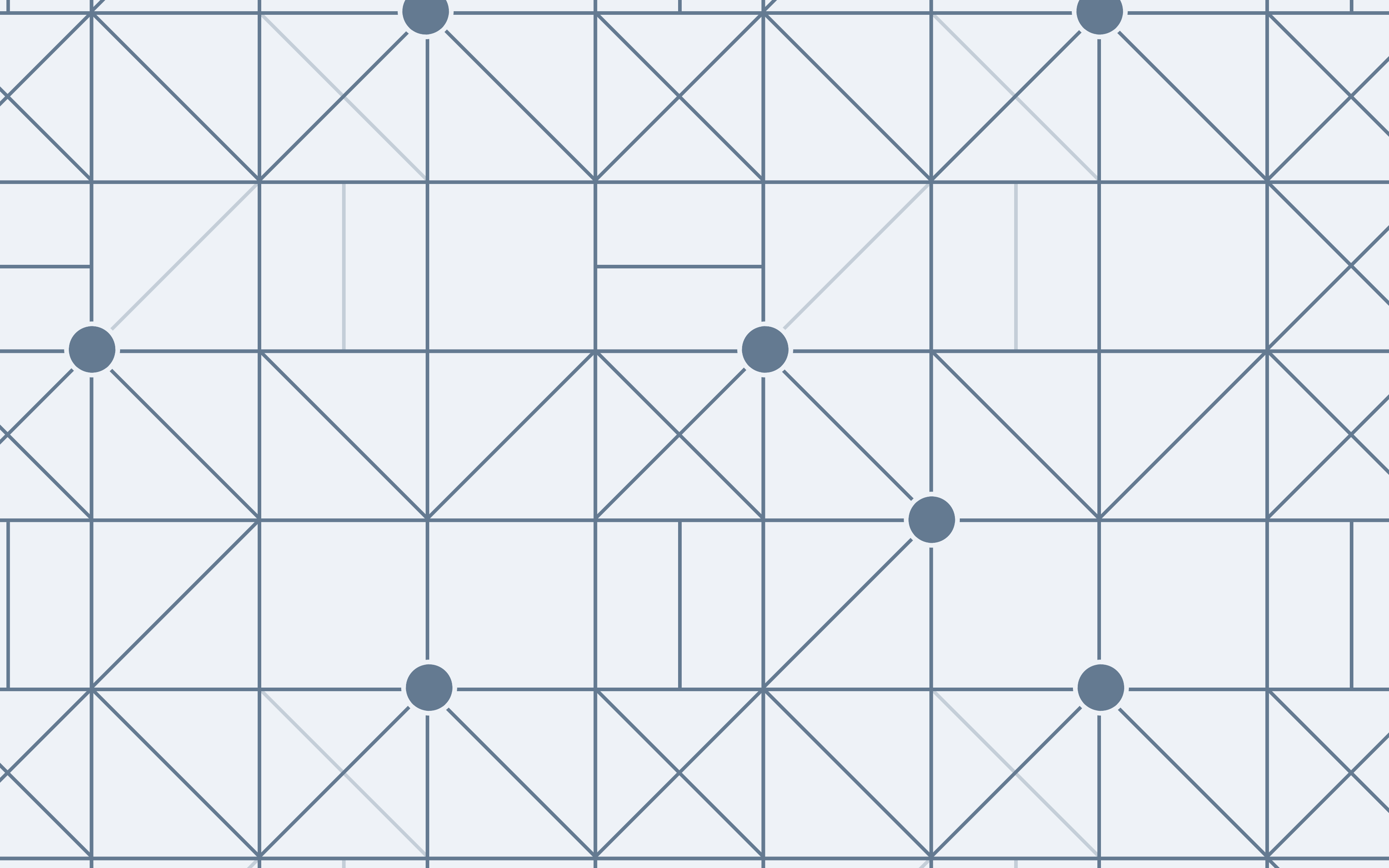


Binary Search: {Searching Algorithm}

TC : $O(\log_2 N)$ SC : $O(1)$

- define range of search
- divide into two halves
- try to eliminate one and take another

Agenda.

LC: Easy

✓ Basic of BS, BS over an inc array, and non-dec array ✓

- BS over a rotated sorted array.
- Bs over a mountain array

} LC: Med

- BS over solu^n

} LC: Hards

Search Pivot in a rotated sorted array.

int[] arr =

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

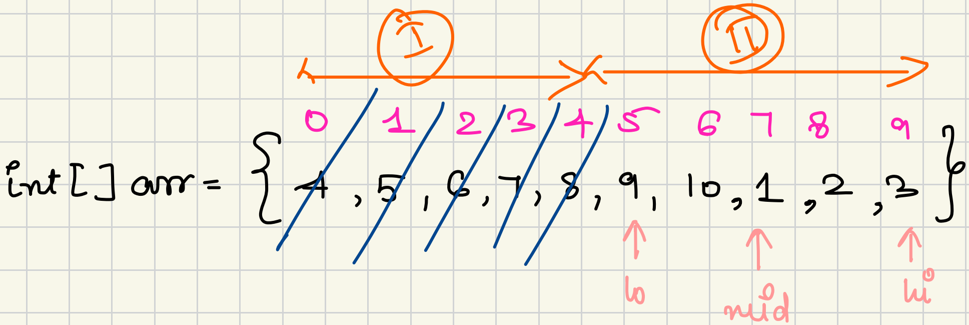
pivot

min^m value in a rotated sorted array

Bruteforce

- Linear search
  - find Min^m value

    TC: O(N) SC: O(1)



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

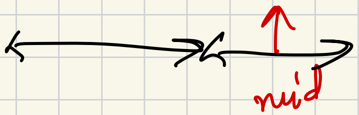
```
if (arr[mid-1] > arr[mid])
        mid is pivot.

else if (arr[mid+1] < arr[mid])
        mid+1 is pivot.
```

discard sorted part and move towards unsorted part.

arr[] =

0	1	2	3	4	5	6	7	8	9	10
~~7~~	~~8~~	~~9~~	~~10~~	~~11~~	~~12~~	~~13~~	~~14~~	~~15~~	16	2
									↑	↑
									lo	hi

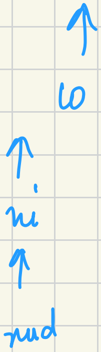
Sorted region



$$\boxed{arr[lo] <= arr[mid]}$$

TC: $O(\log_2 N)$ SC: $O(1)$

$$\begin{array}{ccccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ arr = \{ & \cancel{4}, & \cancel{5}, & \cancel{6}, & \cancel{7}, & 8, & \cancel{1}, & \cancel{2}, & \cancel{3} \} \end{array}$$



pans = ~~4~~ 1

- divide region into 2 half.
- store min value in pans af sorted region to be eliminated

Search an element in a rotated sorted array.

$$\begin{array}{cccccccccc} 0&1&2&3&4&5&6&7&8&9\\ \cancel{4}&\cancel{5}&\cancel{6}&7&\cancel{8}&\cancel{9}&\cancel{10}&\cancel{1}&\cancel{2}&\cancel{3}\end{array}$$

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

target = 7

↑
hi
↑
lo

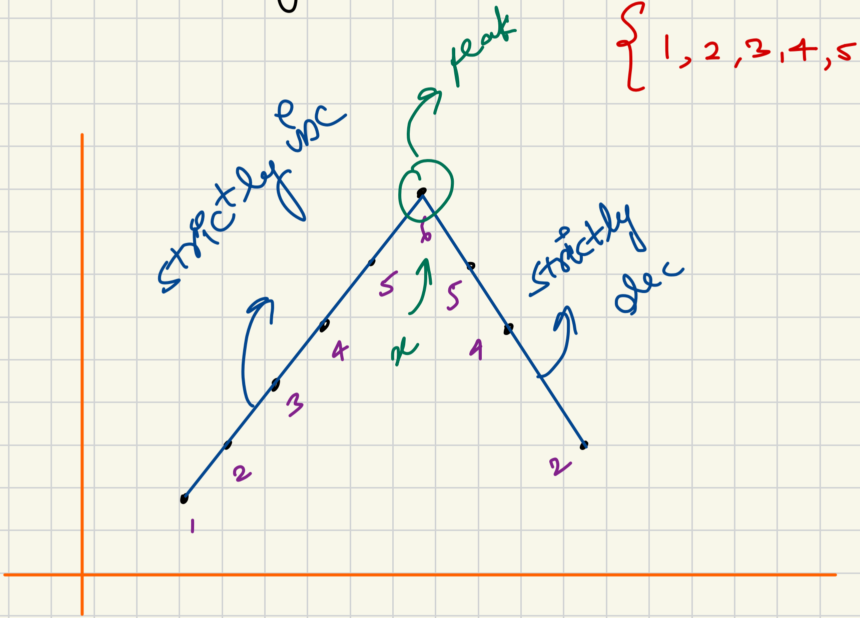
if (arr[mid] == target) return mid;

↑
mid

$arr[lo] \leq x \leq arr[mid]$

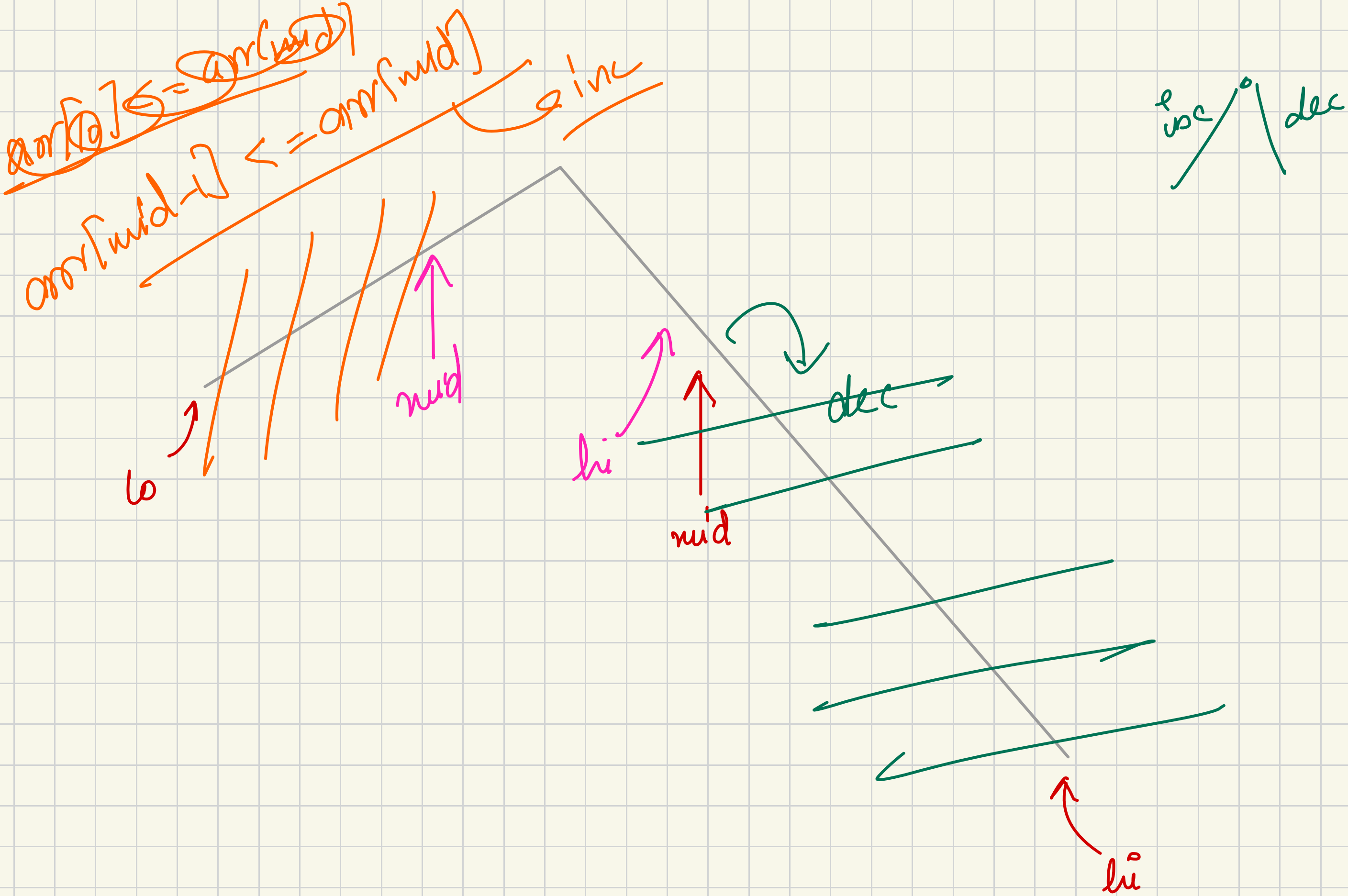
Peak in a Mountain Array.

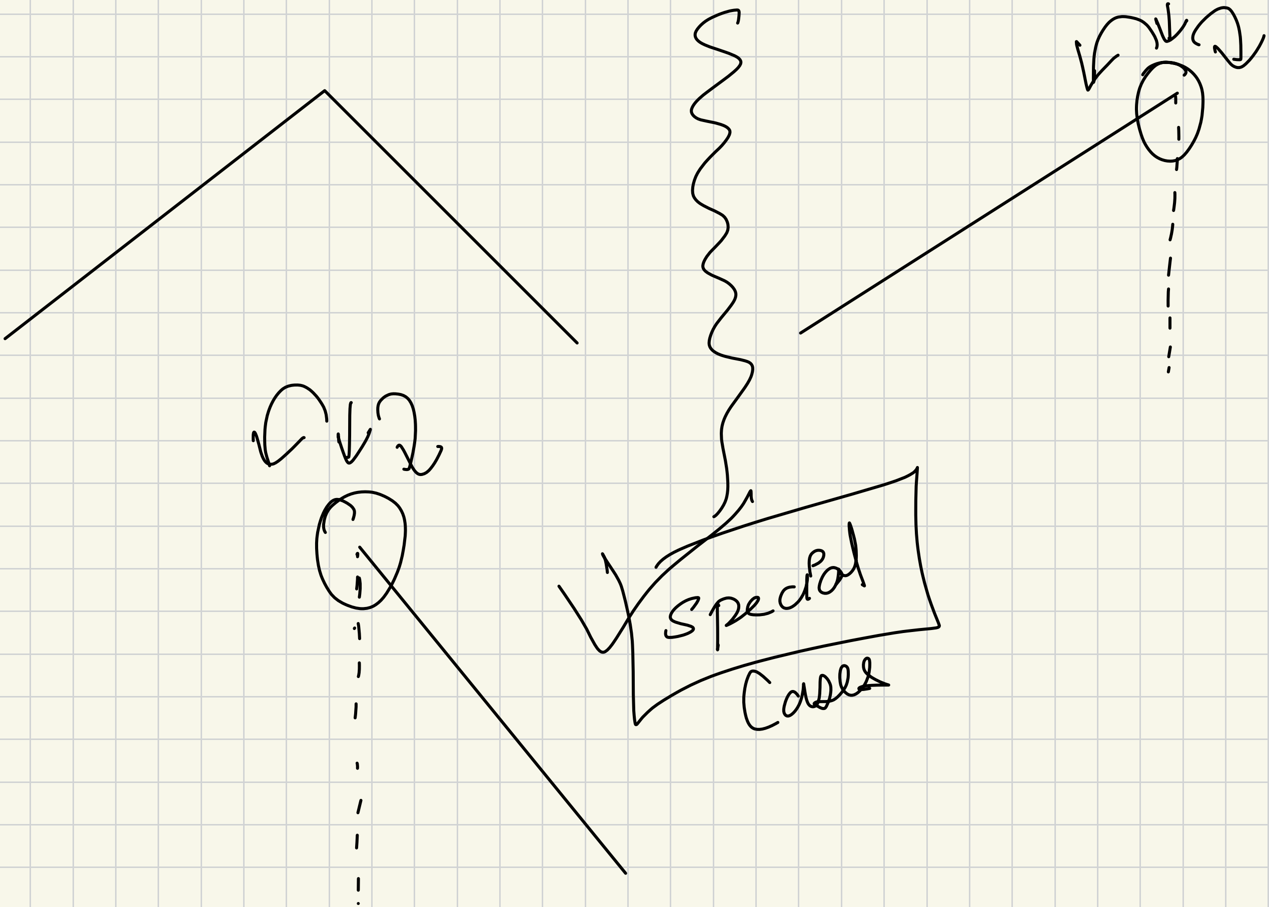
{1, 2, 3, 4, 5, 6, 5, 4, 2}



✓ x^{th} index is peak

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


arr[0] <= arr[mid]
arr[mid-1] <= arr[mid]
inc
lo
mid
hi
mid
dec
hi
inc
dec


special
Cases

Allocate Min^{m} number of Pages

0 1 2 3

int[] Books = {34, 12, 67, 90}

students = 2

- distribute these N Books among M students
- each student gets min^{m} one Book.
- distribution should be contegeneous

int[] Books = $\{\overset{0}{34}, \overset{1}{12}, \overset{2}{67}, \overset{3}{90}\}$

students = 2

way 1

S1 → 34 + 12 + 67 → 113 pages

S2 → 90 → 90 pages

way 2

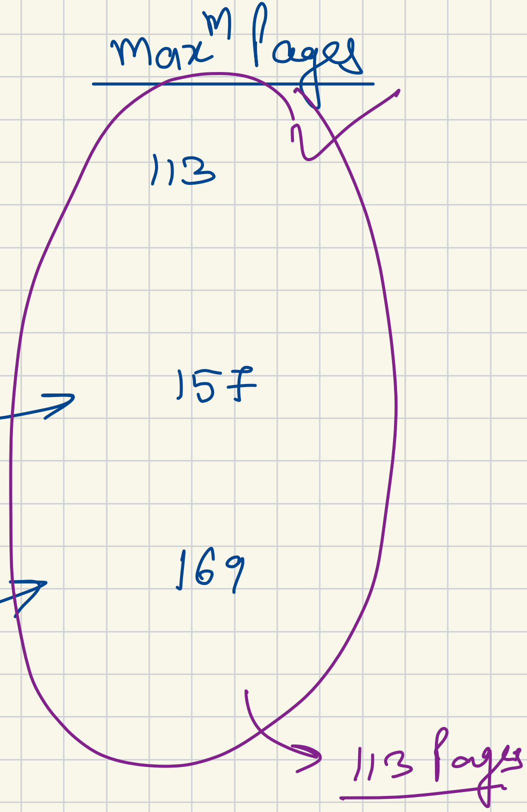
S1 → 34+12 → 46 pages

S2 → 67+90 → 157 pages

way 3

S1 → 34 → 34 pages

S2 → 12+67+90 → 169 pages



Max^m Pages
113
157
169
113 Pages