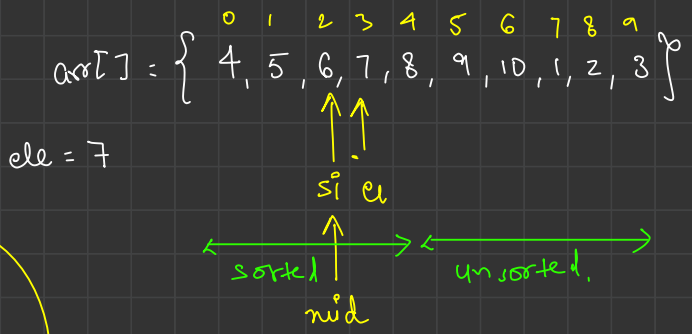
# Search in a rotated sorted array




[4, 8]
↓
Range

```
if (arr[mid] == ele)
    return mid;

else if (arr[si] <= arr[mid])
{
    if (ele in Left Range)
        move left;
    else
        move right;
}
else
{
    if (ele in Right Range)
        move right;
```

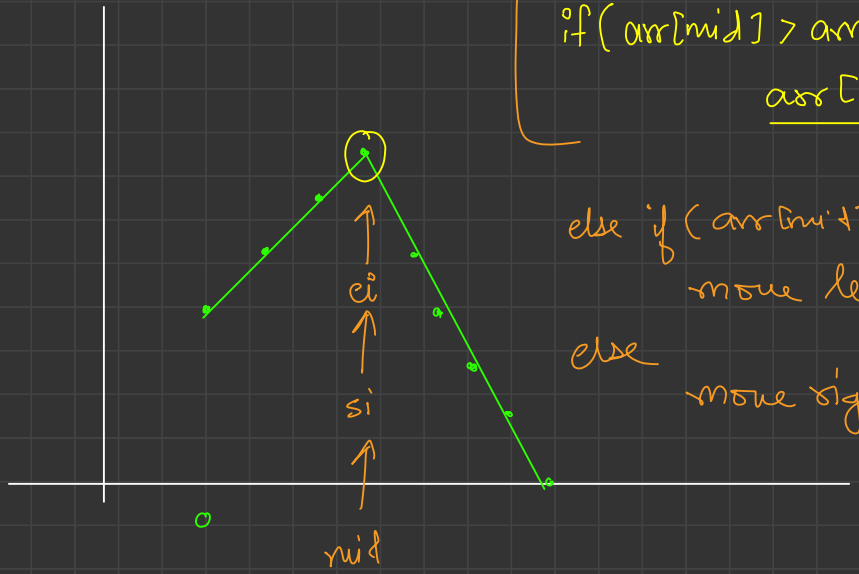
else

move left;

}

# Peak in a Mountain Array

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



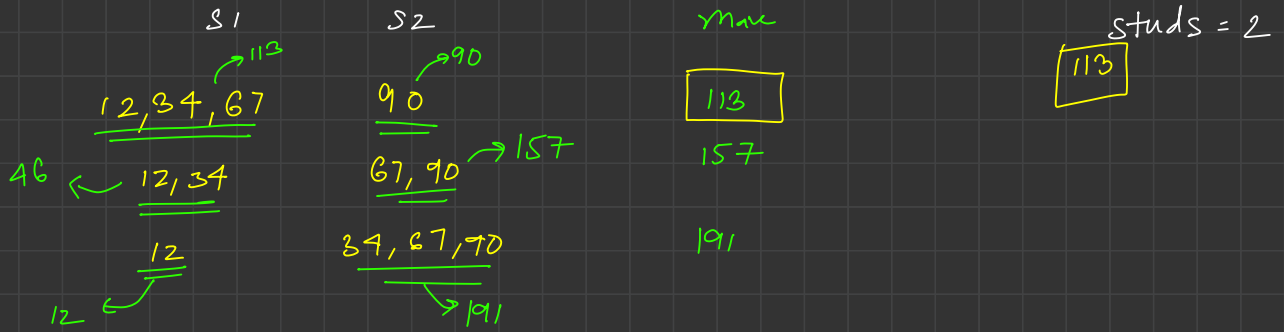

```
if(arr[mid] > arr[mid-1] && arr[mid] > arr[mid+1])
        arr[mid] is peak!

else if(arr[mid] >= arr[mid+1])
        move left;
else
        move right;
```

# Binary Search Over Solution

## Allocate Minm Pages

books[] = {12, 34, 67, 90}  studs = 2




studs = 2

113

books[] = {12, 34, 67, 90}  studs = 2

stud = 1

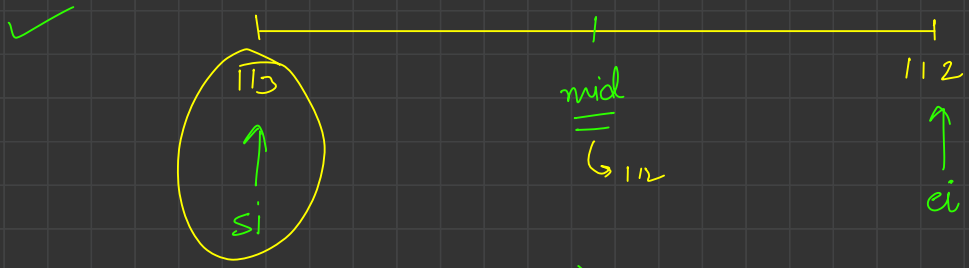
{12, 34, 67, 90}

= 203 pages

stud = 4

way {12} {34} {67} {90}

= 90 pages




pans = ~~146~~ ~~117~~ 113

maxPages = 146

books[] = {12, 34, 67, 90}

S1 → 12 + 34 + 67

S2 → 90

maxPages = 103

books[] = {12, 34, 67, 90}

S1 → 12 + 34

S2 → 67

S3 → 90

maxPages = 117

books[] = {12, 34, 67, 90}

S1 → 12 + 34 + 67

S2 → 90

maxPages = 110

books[] = {12, 34, 67, 90}

S1 → 12 + 34

S2 → 67

S3 → 90

maxPages = 113

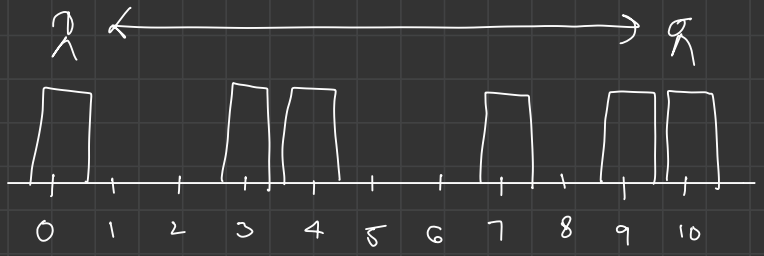
books[] = {12, 34, 67, 90}

S1 → 12 + 34 + 67

S2 → 90

# Aggressive Cows

stalls = {0, 3, 4, 7, 9, 10}  cows = 4




max dist = dist b/w first and last stall

min dist = min dist b/o any two cons stalls


stalls = { 0, 3, 4, 7, 9, 10 }

minDist = 4

ans = ~~2~~ 3