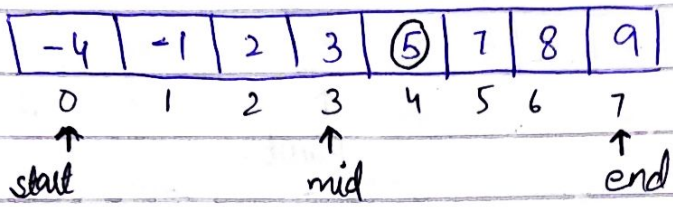
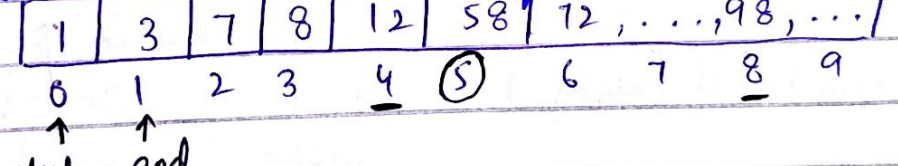
- **Binary Search :-**
- **Algorithm :-**




```
while (start <= mid){
    mid = (start + end) / 2
    if (array[mid] == element) return mid;
    else if (array[mid] > element){
        end = mid - 1;
    else - if (array[mid] < element){
        start = mid + 1;
}
```

- **Search an element in an infinite array :-**

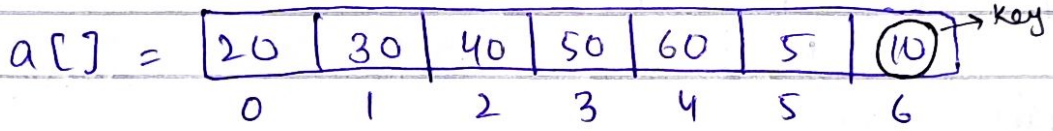



```
while (a[end] < element){
    start = end;  → 1 → 2 → (4) }
    end = end * 2; → 2 → 4 → (8) } → Range
}
return binary_search(array, start, end, element);
```

- In this case we are finding range so that we can solve it in $O(\log N)$.

• Search an Element in a Sorted Rotate Array :-

- 120 (130) 40 50 90 100 110
  ← → ↑ — X —
- 50 60 90 100 20 (30) 40
  — X — ↑ ← →
- (20) 30 40 50 60 5 10
  ← → ↑ — X —
  Taking this part Discarding this part.




```
function modifiedBinarySearch (array, start, end, element)
{
    if (start > end) return -1;
    let mid = floor (start + end / 2);
    if (a[mi] == element) return mid;
    else-if (array [mid] > array [start]) {
        if (element <= a[start] && element < a[mid])
            return modifiedBinarySearch (array, start, mid-1, el);
        else → // it will executed when element is not in sorted array.
            return modifiedBinarySearch (array, mid+1, end, el);
    else-if (array [mid] < array [end]) {
        if (element > mid && element <= a[end]) {
            return modified Array (array, mid+1, end, el)
        else
            return modifiedBinarySea (array, start, mid-1, el);
```

- **Books Allocation :-**

Minimize the maximum pages read by a student.

↓ students.

10	20	5	15	5

∴ K = 2

↑ → This will be the answer

1st → 10	(30)	35	50
2nd → 45	25	20	5

10	10	20	30

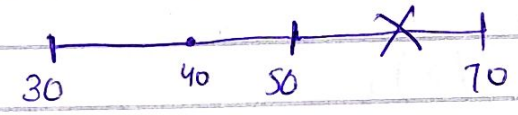
K = 2

min = maxValue = 30

max = SumValues = 70

mid = (min + max)/2 = 50

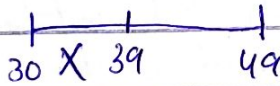
Range



min = 30

max = 49

mid = 39



In this case [10, 10, 20, 30] ∴ K = 3

(10 ↔ 10 : 1, 20 ↔ 2, 30 ↔ 3)

which is not solution

Range

→ Check code in github.