BINARY SEARCH

☆ Binary Search is a searching algo which always only works on sorted arrays or any limited search space that is sorted.
How it works is that it starts by looking at the middle element If it is not equal to our middle element, we check if it will be behind it or ahead it (as our search space is sorted).
So for the next step our search step is halfed, as our element will only be on one side of the middle element and the other side is irrelevant. This is repeated until we find the element or the array runs out in which case we say element not found. Time Comple-xity is $O(\log N)$

Binary Search has two approaches, iterative and recursive

Iterative Pseudocode :

```
binarySearch (arr, N, a) {
    low = 0
    high = N - 1
    mid = (low + high) / 2
    while (low < high) {
```

```
        if (arr[mid] == a) {
            return a
        } else if (arr[mid] > a) {
            low = mid + 1
            mid = (low + high) | 2
        } else {
            high = mid - 1
            mid = (low + high) | 2
        }
    }
    return -1
}
```

Recursive Pseudocode :

```
binary Search (arr, low, high, a) {
    mid = (low + high) | 2
    if (low > high) return -1
    if (arr[mid] == a) {
        return mid
    } else if (arr[mid] > a) {
        return binary Search (arr, mid + 1, high,
                              a)
    } else {
        return binary Search (arr, low, mid - 1, a)
    }
}
```