Binary search

A

4	8	10	15	18	21	24	27	29	33	34	37	39	41	43
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

$\boxed{\text{Let, Key} = 18}$

1st

low $= 0$, high $= 14$, mid $= \dfrac{l+h}{2} = 7$

18 is on the L.H.S of array

$\therefore$ Shift high to (mid-1)

2nd

Low $= 0$, high $=$ mid$-1 = 6$ mid $= 3$

18 is on the R.H.S of array

$\therefore$ Shift to ~~high~~ low to (mid+1)

3rd

Low $= 4$, high $= 6$, mid $= 5$

18 is on the L.H.S of Index 5

$\therefore$ Shift high to (mid-1)

4th

low = 4, high = 4, mid = $\frac{4+4}{2}$ = 4

Is the 4th index element = key = 18?

→ YES | $\therefore$ FOUND

* ONCE low > high (i.e. low should be on L.H.S and high should be in R.H.S, but if low is in R.H.S and high is in L.H.S, that means element is not in the list).

Algorithm for Binary Search

// Iterative version

```
BinSearch(l, h, key)
{
    while(l<=h){
        mid = ⌊(l+h)/2⌋;

        if (key == A[mid])
            return mid;

        else if (key < A[mid])
            h = mid-1;
```

```
        else
            l = mid + 1;
    }
    return -1;
}
```

~~return 0;~~

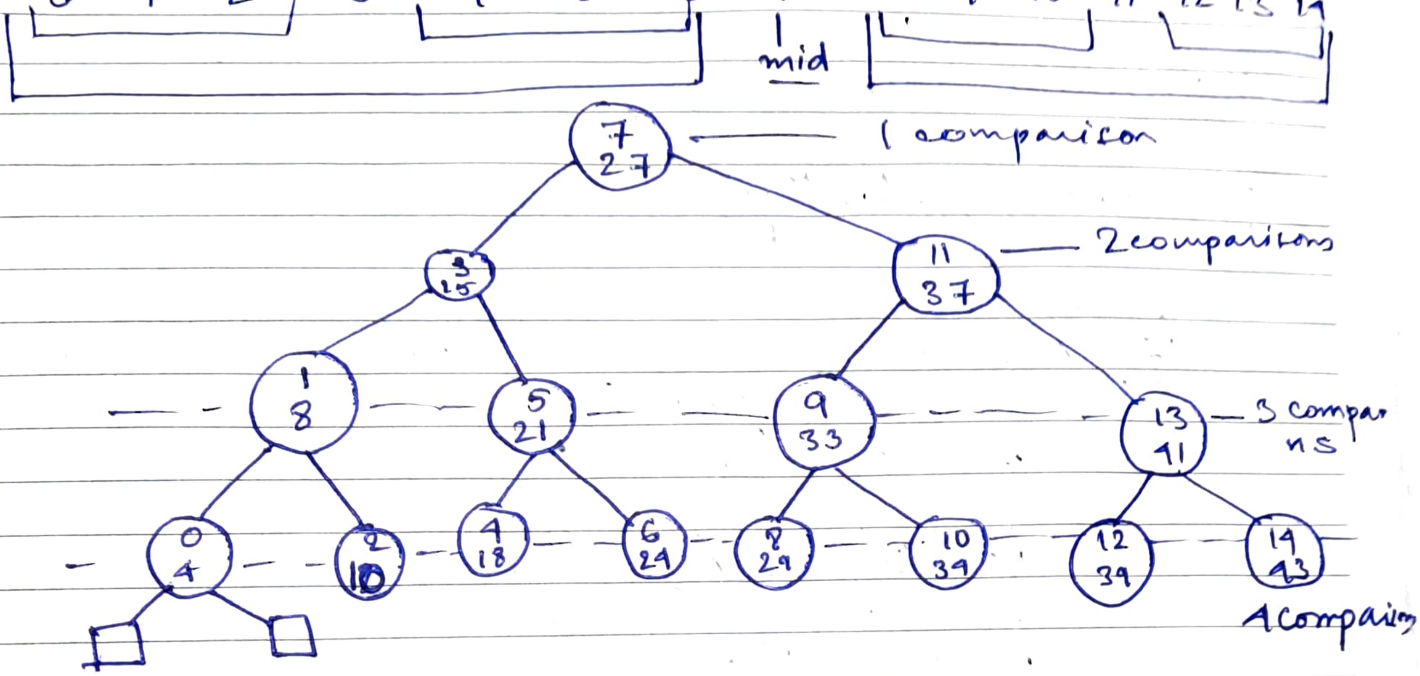
// Recursive Version

```
RBinSearch(l, h, key)
{
    if (l <= h)
    {
        mid = ⌊(l+h)/2⌋;   → take floor value
        if (key == A[mid])
            return mid;
        else if (key < A[mid])
            return RBinSearch(l, mid-1, key);
        else
            return RBinSearch(mid+1, h, key);
    }
    return -1;   // key not found
}
```

Analysis of Binary Search

Array ADT. [Size = 15
length = 15]

4	8	10	15	18	21	24	27	29	33	34	37	39	41	43
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

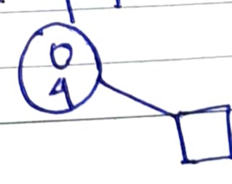


- Comparison depends on height of the tree

∴ Time complexity of binary search is $O(\log n)$

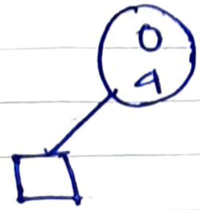
Best case : min : $O(1)$
Worst case : max : $O(\log n)$

* What if I want to make a search that is < 8 but > 4,
it ends up in the square box on the R.H.S of 4



What if I want to make a search that is less than 4_8?

→ It ends up in the square box on the L.H.S of 4



∴ Greater ⟶ R.H.S
Smaller ⟶ L.H.S

Average Case Analysis

$$1 + 1\times2 + 2\times4 + 3\times8$$

1 comparison for 1 element

1 comparisons for 2 elements (each).

2 comparison for 4 elements (each)

3 comparisons for 8 elements (each)

$$= 1 + 1\times2^1 + 2\times2^2 + 3\times2^3$$

$$= \frac{\sum_{i=1}^{\log n} i\times2^i}{n} = \frac{\log n \times 2^{\log n}}{n} = \frac{\log n \times n^{\log 2}}{n}$$

$$= \log n$$