

## Binary Search

A

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

Let, Key = 18

1<sup>st</sup>

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

18 is on the L.H.S of array  
 $\therefore$  Shift high to (mid - 1)

@ 2<sup>nd</sup>

$$\text{low} = 0, \text{ high} = \text{mid} - 1 = 6, \text{ mid} = 3$$

18 is on the R.H.S of array  
 $\therefore$  Shift low to (mid + 1)

@ 3<sup>rd</sup>

$$\text{low} = 4, \text{ high} = 6, \text{ mid} = 5$$

05

MARCH  
Tuesday

65/301

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

10/10/20

10/10/20

8  $\therefore$  Shift high to  $(mid - 1)$ 9 4<sup>th</sup>10 low = 1, high = 4, mid =  $\frac{1+4}{2} = 2$ 11 Is the 4<sup>th</sup> index element = Key = 18?12  $\rightarrow$  YES |  $\therefore$  FOUND

13

14 \* 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)

17 Algorithm for Binary Search, // Iterative version

18 BinSearch (L, h, Key)

19 { while (L <= h) {

mid =  $\lfloor (L+h)/2 \rfloor$ ;

20 if (Key == A[mid])

\* return mid;

else if (Key < A[mid])

h = mid - 1;

Monday	1	8	15	22	29
Tuesday	2	9	16	23	30
Wednesday	3	10	17	24	-
Thursday	4	11	18	25	-
Friday	5	12	19	26	-
Saturday	6	13	20	27	-
Sunday	7	14	21	28	-

10. week

else

 $l = mid + 1;$ 

}

return -1;

}

~~return 0;~~

// Recursive Version

RBinSearch(l, h, Key)

{

if ( $l \leq h$ )

{

 $mid = \left\lfloor \frac{l+h}{2} \right\rfloor;$ 

take floor value

if ( $key == A[mid]$ )

return mid;

else if ( $key < A[mid]$ )

return RBinSearch(l, mid-1, Key);

else  
return RBinSearch(~~l~~ mid+1, h, Key);

} return -1; // Key not found

\*

}



07

MARCH  
Thursday

67/299

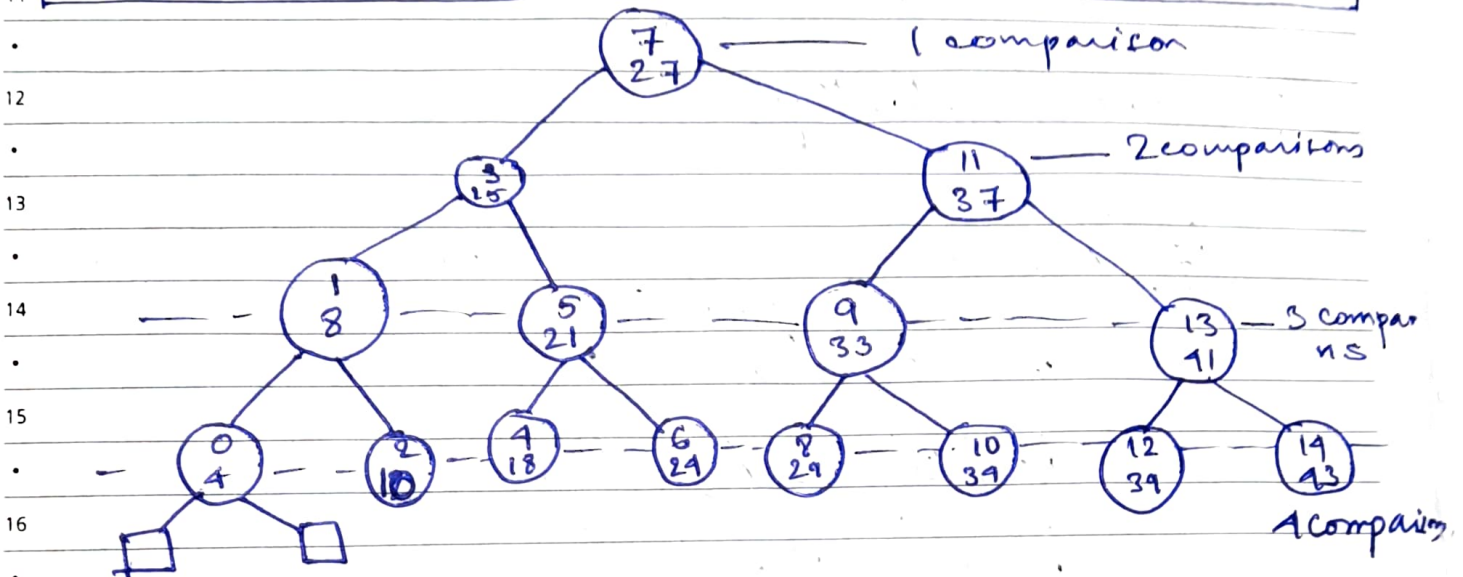
Analysis of Binary Search

10 week

10 week

8 Array ADT. [ Size = 15  
length = 15 ]

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



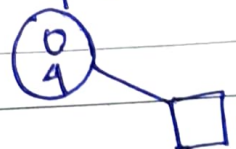
• Comparison depends on height of the tree

∴ 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



You can't turn back the clock, But you can wind it up again. — Bonnie Prudden

Monday	1	8	15	22	29
Tuesday	2	9	16	23	30
Wednesday	3	10	17	24	-
Thursday	4	11	18	25	-
Friday	5	12	19	26	-
Saturday	6	13	20	27	-
Sunday	7	14	21	28	-

MARCH  
Friday

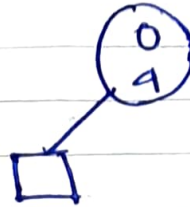
08

6/2/98

10. week

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

→ 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

$$1 + 1 \times 2 + 2 \times 4 + 3 \times 8$$

→ 1 comparison  
for 1 element

1 comparison  
for 2 elements  
(each)

2 comparisons  
for 4 elements  
(each)

3 comparisons  
for 8 elements  
(each)

$$1 + 1 \times 2^1 + 2 \times 2^2 + 3 \times 2^3$$

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

$$\frac{\log n \times 2^{\log n}}{n}$$

$$\frac{\log n \times n^{\log 2}}{n}$$

$$= \log n$$