

BINARY SEARCH

PSEUDO CODE:-

Binary Search (A, Key):

1. first = 1
2. last = length(A)
3. flag = 0
4. while first \leq last and flag == 0
5. mid = (first + last) // 2
6. if A[mid] == key
7. flag = 1
8. elseif key < A[mid]
9. last = mid - 1
10. else
11. first = mid + 1
12. return flag



• DRY RUNNING

BEST CASE:-

$A = [1, 2, 3, 4, 5, 6, 7, 8]$

Key = 4

- In binary search, when key is present at mid then it is best case because it takes only one comparison to find the key

1. first = 1

2. last = 8

3. flag = 0 1st iteration of while

4. While $1 \leq 8$ and $0 == 0$

5. $mid = (1 + 8) // 2 = 4$

6. If $A[4] == 4$ (1)
 $4 == 4$

7. flag = 1

8. _____

9. _____

10. _____

11. _____

2nd iteration of while

4. While $1 \leq 8$ and $1 == 0 \rightarrow$ false
 loop terminates

12. return 1

Line no of code	Time/ Instruction	frequency
1	C	1
2	C	1
3	C	1
4	C	$1+1=2$
5	C	1
6	C	1
7	C	1
8	C	0
9	C	0
10	C	0
11	C	0
12	C	1

$$T(n) = 1+1+1+2+1+1+1+1$$

$$T(n) = 9$$

$$T(n) = K$$

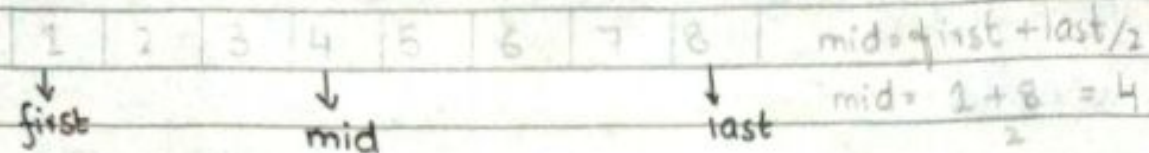
• **DISCUSSION:-**

- Best case time complexity of binary search is $O(1)$
- means that, only takes one comparison to find the key
- Binary search in best case requires constant time to execute

TIME COMPLEXITY:-

Key = 10

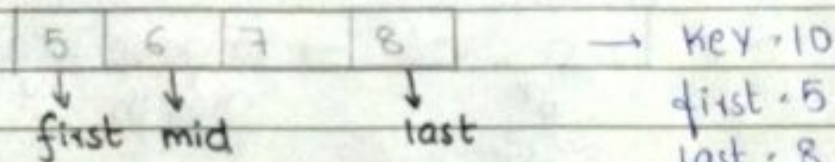
• First iteration



• since $A[mid] < \text{key} \rightarrow$ line no 10
 $4 < 10$

11. $first = mid + 1$
 $\rightarrow first = 5$

• 2nd iteration

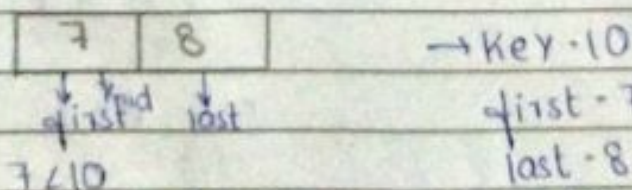


10. since $A[mid] < \text{key}$
 $6 < 10$

mid = $\frac{5 + 8}{2} = 6$

11. $first = mid + 1$
 $\rightarrow 7$

• 3rd iteration :-



10. $A[mid] < \text{key}$

mid = $\frac{7 + 8}{2} = \frac{15}{2} = 7$

11. $first = mid + 1$
 $\rightarrow 7 + 1 = 8$

• 4th iteration



1st iteration - $\{1, 2, 3, 4, 5, 6, 7, 8\} \rightarrow n \rightarrow \frac{n}{2^0} = n$

2nd iteration - $\{5, 6, 7, 8\} \rightarrow n/2 \rightarrow \frac{n}{2^1} = \frac{n}{2}$

3rd iteration - $\{7, 8\} \rightarrow n/4 \rightarrow \frac{n}{2^2} = \frac{n}{4}$

4th iteration - $8 \rightarrow n/8 \rightarrow \frac{n}{2^3} = \frac{n}{8}$

5th iteration - doesn't run

In general terms

$\frac{n}{2^k} = 1 \rightarrow$ To prove logarithmic growth

amount of steps to make $n=1$

8, 4, 2, 1

Indicates logarithmic growth

• If $n=16$

$$\frac{16}{2^k} = 1$$

$$16 = 2^k$$

$$2^4 = 2^k$$

$$k=4 \rightarrow n=16$$

• If $n=4$

$$\frac{4}{2^k} = 1$$

$$4 = 2^k$$

$$4 = 2^2$$

$$2^2 = 2^k$$

$$k=2 \rightarrow n=4$$

• If $n=8$

$$\frac{8}{2^k} = 1$$

$$8 = 2^k$$

$$8 = 2^3$$

$$2^3 = 2^k$$

$$k=3 \rightarrow n=8$$

Line no of code	Time/ Instruction	frequency
4	C	$5 = \log_2(n) + 2 = \log_2(6) + 2 = 5$ ↳ express in terms of log because we prove that the growth is logarithmic
5	C	$4 = \log_2(n) + 1$
6	C	$4 = \log_2(n) + 1$
7	C	0
8	C	$4 = \log_2(n) + 1$
9	C	0
10	C	$4 = \log_2(n) + 1$
11	C	$4 = \log_2(n) + 1$
12	C	1

$$t(n) = k \log_2(n) + k$$



$$T(n) = \log_2(n) + 2 + \log_2(n) + 1 + \log_2(n) + 1 + \log_2(n) + 1 + \log_2(n) + 1 + \log_2(n) + 1$$

$$T(n) = 6 \log_2(n) + 6$$

$$T(n) = k \log_2(n) + k$$

• Discussion:-

- Worst case complexity of binary search is $O(\log n)$
- binary search in worst case grows logarithmically.