

(op hand)

noon

Binary Search

1 pm

→ Optimised way of searching in array

2 pm

→ Arrays need to be sorted

3 pm

in Linear Search : $N \Rightarrow$ no. of elements max. comparison

4 pm

5 pm

arr: $[2, 4, 6, 9, \underbrace{11}_{\text{mid}}, \overbrace{12}^{s_1}, \overbrace{14}^{s_1}, \underbrace{20}_{m_1}, \overbrace{36}^{s_1}, \overbrace{48}^e]$ target = 36

evening

step-1 = Find middle element

7 pm

step-2 : If target > mid \Rightarrow search in right

else search in left

3 : If middle == target, answer

4 : If start > end : element not found

④ → we are not creating new array

Dry run

8 am S

2 4 6 9 11 12 14 20 36 48

9 am

1 2 3 4 5 6 7 8 9 10

10 am

$$m = \frac{s + e}{2} = \frac{1 + 10}{2} = 5$$

11 am

36 > 11

S m e
12, 14, 20, 36, 48
6 7 8 9 10

noon

1 pm

$$m = \frac{s + e}{2} = \frac{1 + 5}{2} = 3$$

2 pm

36 > 20

S m e
36, 48
9 10

3 pm

target

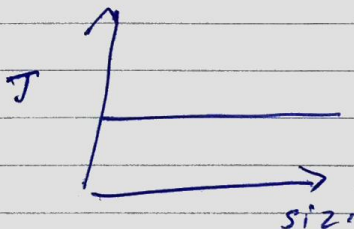
4 pm

Comparison in best case:-

5 pm

$O(1)$

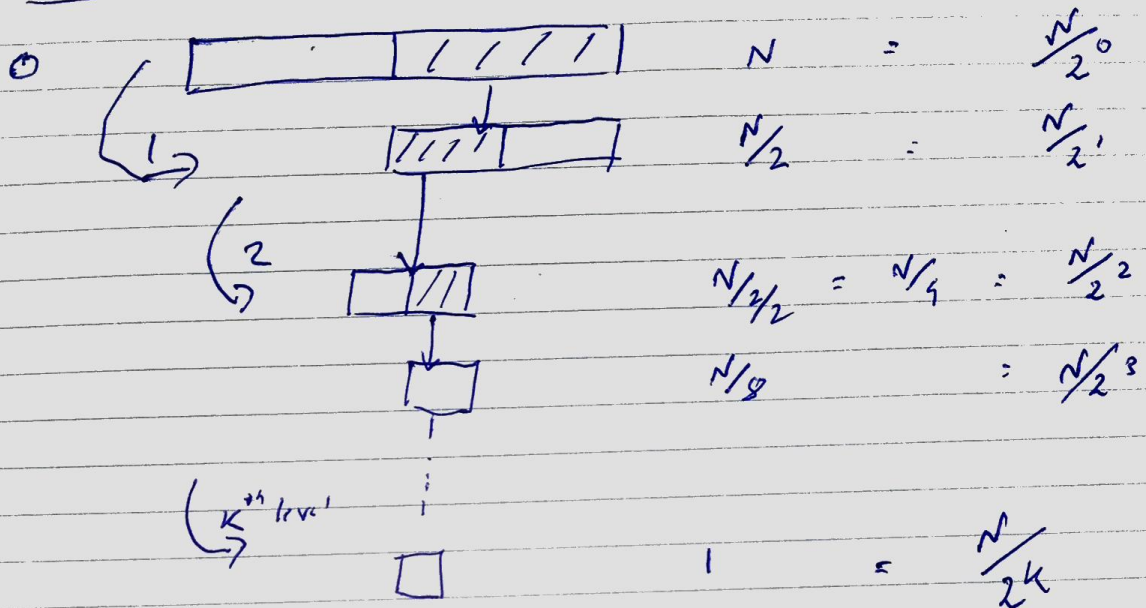
evening



7 pm

FEB' 21

Worst case



Also,

$$\frac{N}{2^k} = 1 \Rightarrow N = 2^k \Rightarrow \log(N) = \log(2^k)$$

$$\Rightarrow \log N = k \log 2 \Rightarrow k = \frac{\log N}{\log 2}$$

Sunday 24

$$k = \log_2 N$$

$$\text{Worst case comparison} = \underline{\log_2 N} = O(\log N)$$

$$\therefore \text{Search in } 1000000 \Rightarrow \text{Linear} = 1000000$$

$$\text{Binary} = 20$$

JANUARY 2021

25

MONDAY

05 Wk / 025-340

M	T	W	T	F	S	S	M	T	W	T	F	S
					1	2	3	4	5	6	7	8
11	12	13	14	15	16	17	18	19	20	21	22	23
25	26	27	28	29	30	31						

JAN 21

Order Agnostic Binary Search:-

8 am

9 am

→ If we don't know the order is in ascending or descending but the array is sorted

10 am

11 am

arr: [3, 3, 3, 4, 12, 60, 80, 120, 200]

Problem

S E

noon

1 pm

compare: S and E

2 pm

$S > E \rightarrow$ ascending

3 pm

$E > S \rightarrow$ descending

4 pm