

Date 17 Oct 2023

Page \_\_\_\_\_

Day - 30

## Binary Search

=> Binary search is real world based searching method.

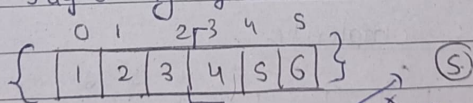
=> Just like we do in dictionary for finding a word, same we do in binary search.

=> Here, the main condition for BS is that the array should be sorted.

=> First we find the mid of the array then check the element at mid with the element we want to search.

=> If it is that element then you have found the element otherwise check it is greater or lesser than the mid element.

=> And go on that side as left if lesser and right if greater.



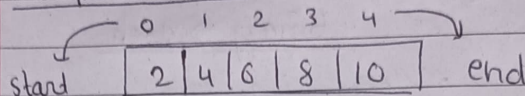
=>  $\text{mid} = \frac{0+5}{2} = 2 \rightarrow 3$

=>  $\text{mid} = \frac{3+5}{2} = 4 \rightarrow 5$   
found

=> ~~find~~ the element.

Date \_\_\_\_\_

Page \_\_\_\_\_

Implementation

key = 4

$$\textcircled{1} \quad \text{mid} = \frac{\text{start} + \text{end}}{2}$$

$\textcircled{2} \quad \text{if (arr[mid] == key)}$   
           cout << "Got it" stop,

else if (arr[mid] < key)

Right side (start = mid + 1;

else

Left side (end = mid - 1);

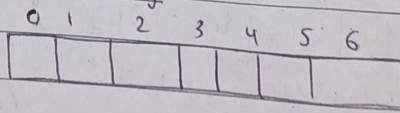
⇒

condition for breaking the loop:  
 while (start <= end) {

  |  
 }

Time Complexity

⇒

 $\textcircled{1}$ 

$$N \rightarrow 1$$

 $\textcircled{2}$ 

$$N/2 \rightarrow 1$$

 $\textcircled{3}$ 

$$N/4 \rightarrow 1$$

|

$$N/2^k \rightarrow 1$$

⇒

$$N = 2^k \Rightarrow$$

$$\log N = k \log 2$$

$$k = \log_2 N$$



Date \_\_\_\_\_

Page \_\_\_\_\_

=> Worst Case:  $O(\log_2 N)$

=> Best Case:  $\Omega(1)$

=> Worst Case:  $O(\log N)$

=> The formula of mid that we use gives us Integer overflow error.

+ 1;)

$$\text{mid} = \frac{\overset{2^{30}+1}{\text{start}} + \overset{2^{31}-1}{\text{end}}}{2}$$

=>  $\text{mid} = \text{start} + \left( \frac{\text{end} - \text{start}}{2} \right)$

This will not give error.