Binary Search

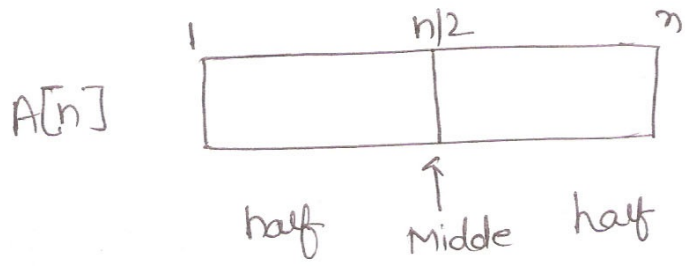
Binary Search is much faster than linear Search.

→ It uses the technique - Divide and Conquer.

→ Array is divided into two half.

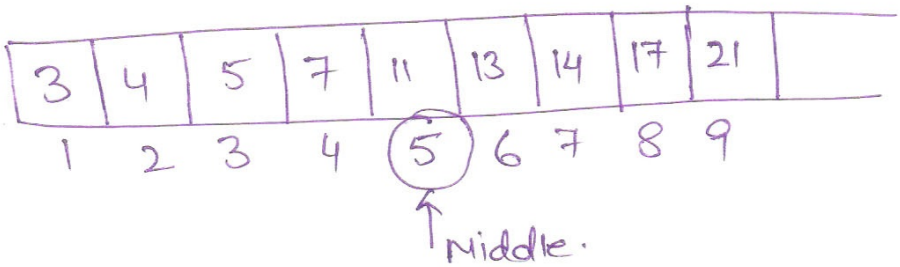
→ Now the desired element is presented in first half or the second half of the array.

→ We search until we d found the element.



** for using Binary Search : Array should be sorted in advance.

first Step



→ Suppose we want to Search Value 14 and its position.

→ We have to find the middle position of array

$$(L_B + U_B)/2 = (9+1)/2 = 5$$

→ Value at 5 location is 11

→ As we are Searching 14 so we will Compare middle Value to our Searching value in order to decide in which half of array we have to move.

11 is less than 14

Mean Search element is in second half that means after 5th element

→ This indicates that searching element will be b/w Series[6] to Series[9]

6	7	8	9
13	14	17	21

↑ Middle

→ Now again we will find middle value

$(6+9)/2 = 7$

→ Compare Series[7] element to 14

14 = 14 here we find our value

So the position is Series[7] = 7th place.

Algorithm

Step: 1. first = 1; Last = N; pos = 0; flag = false

2. While (first <= last and (flag = false) perform step 3 to 5

3. Middle = (first + Last) div 2

4. If (Series[Middle] = ('x') then [Pos = Middle; flag = true]

5. If (Series[Middle] < ('x') then first = Middle + 1;

else Last = Middle - 1;

6. If (flag = true) then print pos

else print 'unsuccessful Search'

7. End.

1	2	3	4	5	6	7	8
6	8	10	13	18	20	27	30

↑ first ↑ middle ↑ Last

(8+1)/2 = 9/2 = 5