Ahaan Desai
23102016
Batch A(1-20)

DAA Lab 2

Algorithm:

I) linear-search (array, element)
// Input: array → Array of integers,
element → integer
// Output: Index of first occurence of element in array (if found) or -1.
n = length (array)
for i = 0 to n-1:
if array[i] == element: ~~then~~
return i
return -1

II) binary-search (array, element, start, end)
// Input: array → Array of integers (sorted)
element → integer
start → integer, default 0
end → integer, default n-1,
where n is length of array.
// Output :- Index of element in array (if found) or -1.
if start > end return -1
int mid = (start + end)/2
if array[mid] == element:
return mid

else if array[mid] > element:
    return binary-search(array, element, start, mid-1)

else
    return binary-search(array, element, mid+1, end)

Testcases:-

I) Linear Search:-

	Array	Element	Expected Output
i)	[1, 5, 4, 2, 3]	4	2
ii)	[10, 7, 15, 203, 51]	16	-1
iii)	[20, 25, 31, 400, 65]	25	1
iv)	[1, 9, -1, -2, -100, -61]	-2	3
v)	[-105, 66, 111, 215, 330]	-60	-1

II) Binary Search:-

	Array	Element	Expected Output
i)	[6, 7, 8, 9, 10]	9	3
ii)	[100, 102, 104, 110, 115]	115	4
iii)	[21, 23, 24, 25, 28]	23	1
iv)	[52, 56, 57, 58, 60, 62]	55	-1
v)	[-4, -3, -2, -1, 0]	-5	-1

Time Complexity :-

i) Linear Search :-

Basic operation is checking if array[i] == element.

$$\text{Time} = \sum_{I=0}^{n-1} (1)$$

$$= (n-1+1)$$

$$= n$$

∴ The time complexity is $O(n)$.

ii) Binary Search :-

In every operation, the element is compared with middle value of array (basic operation) & then the size of the search is reduced by half, if the array element is not equal to the middle value.

Hence,

$$t(n) = 1 + t(n/2)$$

if $n = 2^i$,

$$t(2^i) = 1 + t(2^i)$$

$$= 2 + t(2^{i-1})$$

$$= 3 + t(2^{i-2})$$

$$\vdots$$

$$= i+1 + t(2^0)$$

$\therefore t(2^i) = i+1 + t(1)$

For array of size 1, let the time taken be 1

$\therefore t(2^i) = i+2$

Now, $2^i = n \rightarrow i = \log_2 n$

$\therefore t(n) = \log_2 n + 2$

$\therefore t(n) = O(\log n)$

Hence the time complexity of binary search is $O(\log n)$.