COURSE HANDOUT

Course Code	ACSC13
Course Name	Design and Analysis of Algorithms
Class / Semester	IV SEM
Section	A-SECTION
Name of the Department	CSE-CYBER SECURITY
Employee ID	IARE11023
Employee Name	Dr K RAJENDRA PRASAD
Topic Covered	Binary Search Algorithm
Course Outcome/s	Apply binary search for determining whether key element is found or not in a given array of elements.
Handout Number	16
Date	17 April, 2023

Content about topic covered: Binary Search Algorithm

Binary Search Algorithm (Recursive)

```
Algorithm BinSrch(a, i, l, x)
// Given an array a[i : l] of elements in nondecreasing
// order, 1 ≤ i ≤ l, determine whether x is present, and
// if so, return j such that x = a[j]; else return 0.
{
    if (l = i) then  // If Small(P)
    {
        if (x = a[i]) then return i;
        else return 0;
    }
    else
    { // Reduce P into a smaller subproblem.
        mid := ⌊(i + l)/2⌋;
        if (x = a[mid]) then return mid;
        else if (x < a[mid]) then
                 return BinSrch(a, i, mid − 1, x);
             else return BinSrch(a, mid + 1, l, x);
    }
}
```

Binary Search Algorithm (Iterative)

```
Algorithm BinSearch(a, n, x)
// Given an array a[1 : n] of elements in nondecreasing
// order, n ≥ 0, determine whether x is present, and
// if so, return j such that x = a[j]; else return 0.
{
    low := 1; high := n;
    while (low ≤ high) do
    {
        mid := ⌊(low + high)/2⌋;
        if (x < a[mid]) then high := mid − 1;
        else if (x > a[mid]) then low := mid + 1;
             else return mid;
    }
    return 0;
}
```

Eg: Let us consider the following 14 elements

-15,-6, 0, 7, 9, 23, 54, 82, 101, 112, 125, 131, 142, 151

x = 151

low	high	mid
1	14	7
8	14	11
12	14	13
14	14	14
		Found

x = -14

low	high	mid
1	14	7
1	6	3
1	2	1
2	2	2
2	1	Not found

x = 9

low	high	mid
1	14	7
1	6	3
4	6	5
		Found

Time complexity of successful search = best O(1),
average O(log n),
Worst O(log n).

Time complexity of unsuccessful search= O(log n).