

Lab Manual

Course Title: Design and Analysis of Algorithms

Week 1:

Note: Input, output format for problem I, II and III is same and is given at the end of this exercise.

- I. Given an array of nonnegative integers, design a linear algorithm and implement it using a program to find whether given key element is present in the array or not. Also, find total number of comparisons for each input case. (Time Complexity = $O(n)$, where n is the size of input) **Sample I/O Problem - 1:**

Input: 3 8 34 35 65 31 25 89 64 30 89 5 977 354 244 546 355 244 6 23 64 13 67 43 56 63	Output: Present 6 Present 3 Not Present 6
---	---

- II. Given an already sorted array of positive integers, design an algorithm and implement it using a program to find whether given key element is present in the array or not. Also, find total number of comparisons for each input case. (Time Complexity = $O(n \log n)$, where n is the size of input).
- III. Given an already sorted array of positive integers, design an algorithm and implement it using a program to find whether a given key element is present in the sorted array or not. For an array $arr[n]$, search at the indexes $arr[0]$, $arr[2]$, $arr[4]$, ..., $arr[2^k]$ and so on. Once the interval $(arr[2^k] < key < arr[2^{k+1}])$ is found, perform a linear search operation from the index 2^k to find the element key. (Complexity $< O(n)$, where n is the number of elements need to be scanned for searching):

Jump Search

Input format:

The first line contains number of test cases, T .

For each test case, there will be three input lines.

First line contains n (the size of array).

Second line contains n space-separated integers describing array.

Third line contains the key element that need to be searched in the array.

Output format:

The output will have T number of lines.

For each test case, output will be “**Present**” if the key element is found in the array, otherwise “**Not Present**”.

Also, for each test case output the number of comparisons required to search the key.

Sample I/O Problem - 2, 3:

Input: 3 5 12 23 36 39 41 41 8 21 39 40 45 51 54 68 72 69 10 101 246 438 561 796 896 899 4644 7999 8545 7999	Output: Present 3 Not Present 4 Present 3
---	---