G. H. RAISONI COLLEGE OF ENGG., NAGPUR
(An Autonomous Institute under UGC Act 1956)
Department of Computer Science & Engg.

Date: 29/05/2020

Practical Subject: Design and Analysis of Algorithms
Session: 2020-21

Student Details:

Roll Number	49
Name	Shivam Tawari
Semester	3
Section	A
Branch	Artificial Intelligence

Practical Details: Practical Number- 1

Practical Aim	To study the basic concepts of Linear and Binary Search
Theory & Algorithm	**Linear Search:** ***Theory:*** Linear search is a very basic and simple search algorithm. In Linear search, we search an element or value in a given array by traversing the array from the starting, till the desired element or value is found. It has complexity of O(n) and can be applied in unsorted list too. ***Algorithm:*** Linear Search (List A, Item x) Step 1: START Step 2: Set i to 1 Step 3: if i > n then go to step 7 Step 4: if A[i] = x then go to step 6 Step 5: Set i to i + 1 Step 6: Go to Step 2

	Step 7: Print Element x Found at index i and go to step 8 Step 8: Print element not found Step 9: STOP **Binary Search:** ***Theory:*** Binary search is a search algorithm that finds the position of a target value within a sorted array. Binary search compares the target value to the middle element of the array. If they are not equal, the half in which the target cannot lie is eliminated and the search continues on the remaining half, again taking the middle element to compare to the target value, and repeating this until the target value is found. If the search ends with the remaining half being empty, the target is not in the array. This algorithm has time complexity of O(log(n)) and can only be applied to sorted list. ***Algorithm:*** Step 1: START Step 2: Get element and set L = 0, R = n – 1 Step 3: If L > R, return -1 and goto Step 6 Step 4: Set mid to (L + R)/ 2 Step 5: If arr[mid] < element, then set L = mid +1, Else if arr[mid] > element, right = mid – 1 and goto Step 3, else return mid. Step 6: Print index if result if not equals to -1 else print Not found. Step 7: STOP
Complexity	**Linear Search:** **Best:** $\Omega(1)$ **Average:** $\theta(n)$ **Worst:** $O(n)$ **Binary Search:** **Best:** $\Omega(1)$ **Average:** $\theta(\log(n))$ **Worst:** $O(\log(n))$

Program

Linear Search:

```
def linear_search(myList,item):
    for i in range(len(myList)):
        if myList[i]==item:
            return i
    return -1

n = int(input("Number of elements:"))
myList = []
for a in range(n):
    number = int(input("Enter Element: "))
    myList.append(number)
print("Elements in List :", myList)

x = int(input("Enter element to be Searched: "))

result = linear_search(myList,x)
if result==-1:
     print("Element not found in the list")
else:
     print( "Element " + str(x) + " is found at index %d" %(result))
     print( "Element " + str(x) + " is found at position %d" %(result+1))
```

Binary Search:

```
from math import floor

def binary_search(Array, Search_Term):
    n = len(Array)
    L = 0
    R = n-1

    while L <= R:
        mid = floor((L+R)/2)
        if Array[mid] < Search_Term:
            L = mid + 1
        elif Array[mid] > Search_Term:
            R = mid - 1
        else:
            return mid
    return -1

x = int(input("Number of elements:"))
myList = []
for a in range(x):
    number = int(input("Enter Element: "))
    myList.append(number)
print("Elements in List :", myList)

element=int(input("Element to be searched:"))
index = binary_search(myList, element)
if index >= 0:
    print( "Element " + str(element) + " is found at index %d" %(index))
    print( "Element " + str(element) + " is found at position %d" %(index+1))
else:
    print("Search element not found")
```

Output	**Linear Search:** In [13]: runfile('C:/Users/shiva/untitled1.py', wdir='C:/Users/shiva') Number of elements:4 Enter Element: 11 Enter Element: 22 Enter Element: 33 Enter Element: 44 Elements in List : [11, 22, 33, 44] Enter element to be Searched: 22 Element 22 is found at index 1 Element 22 is found at position 2 **Binary Search:** In [32]: runfile('C:/Users/shiva/untitled1.py', wdir='C:/Users/shiva') Number of elements:4 Enter Element: 12 Enter Element: 24 Enter Element: 36 Enter Element: 48 Elements in List : [12, 24, 36, 48] Element to be searched:36 Element 36 is found at index 2 Element 36 is found at position 3