Modern Education Society's
College of Engineering, Pune

NAME OF STUDENT: Sandesh Santosh Pabitwar	**CLASS: Comp A**
SEMESTER/YEAR: III	**ROLL NO:**
DATE OF PERFORMANCE: 1/12/2021	**DATE OF SUBMISSION: 12/12/2021**
EXAMINED BY: Prof. Amol Kamble	**EXPERIMENT NO: DSL B-11**

TITLE: SEARCHING OPERATIONS

PROBLEM STATEMENT:

a) Write a Python program to store roll numbers of student in array who attended training program in random order. Write function for searching whether particular student attended training program or not, using Linear search and Sentinel search.
b) Write a Python program to store roll numbers of student array who attended training program in sorted order. Write function for searching whether particular student attended training program or not, using Binary search and Fibonacci search

OBJECTIVES:

1. To understand structure of Array.
2. To understand How to search given key using different searching operations.

OUTCOME:

1. To operate on the various structured data.
2. To analyze the problem to apply suitable algorithm and data structure.

PRE-REQUISITES:

1. Knowledge of Python Programming
2. Knowledge of searching methods and array.

APPARATUS:

QUESTIONS: 1. Compare all searching algorithms with its time complexity.(Write answer in tabular format)

```python
def linearSearch(arr, target):
    for i in range(len(arr)):
        if arr[i] == target:
            print('\n\t\tgiven roll number found in arr')
        # else:
        #     print('given roll no. not found in arr')
    if target not in arr:
        print('\n\t\tgiven roll no. not found in arr')

def sentinalSearch(arr, target):
    last = arr[len(arr) - 1]
    i = 0
    arr[len(arr) - 1] = target
    while arr[i] != target:
        i += 1
    arr[len(arr) - 1] = last
    if (i < len(arr) - 1) or (arr[len(arr) - 1] == target):
        print('\n\t\tgiven roll no. found in arr')
    else:
        print("\n\t\tgiven roll no. Not found")


def binarySearch(arr, l, r, target):
    if r >= l:

        mid = l + (r - l) // 2

        # If element is present at the middle itself
        if arr[mid] == target:
            return mid

        # If element is smaller than mid, then it
        # can only be present in left sub-array
        elif arr[mid] > target:
            return binarySearch(arr, l, mid - 1, target)

        # Else the element can only be present
        # in right sub-array
        else:
            return binarySearch(arr, mid + 1, r, target)

    else:
        # Element is not present in the array
        return -1


def fibonacci_search(arr, target):
    size = len(arr)

    start = -1

    f0 = 0
    f1 = 1
    f2 = 1
    while (f2 < size):
        f0 = f1
        f1 = f2
        f2 = f1 + f0
```

```
    while (f2 > 1):
        index = min(start + f0, size - 1)
        if arr[index] < target:
            f2 = f1
            f1 = f0
            f0 = f2 - f1
            start = index
        elif arr[index] > target:
            f2 = f0
            f1 = f1 - f0
            f0 = f2 - f1
        else:
            return index
    if (f1) and (arr[size - 1] == target):
        return size - 1
    return None


############### menu driven program ############################
arr = list(map(int, input('enter roll numbers of students who attended training program\t').split()))
target = int(input('enter roll no. which you want to search: '))
print('by using which search algorithm you want to search\n 1.linear search \n 2.sentinel Search \n 3.binary search '
      '\n 4.fibonacci search ')
ch = int(input('enter choice: '))
if ch == 1:
    linearSearch(arr, target)
elif ch == 2:
    sentinalSearch(arr, target)
elif ch == 3:
    if binarySearch(arr, 0, len(arr) - 1, target) != -1:
        print('\n \t\troll number found in arr')
    else:
        print('\n\t\troll no.not found in arr')
elif ch == 4:
    if fibonacci_search(arr, target) is None:
        print('\n\t\troll no.not found in arr')
    else:
        print('\n\t\troll number found in arr')

```

OUTPUT:

```
enter roll numbers of students who attended training program    1 12 473 8 9
enter roll no. which you want to search: 473
by using which search algorithm you want to search
 1.linear search
 2.sentinel Search
 3.binary search
 4.fibonacci search
enter choice: 4

        roll number found in arr

Process finished with exit code 0
```

Name: Sandesh Santosh Pabitwar Class: CompA

PRN.: F2011040 Sub: FDS lab. B-II.

Q1. Compare all searching algorithms with its time complexity.

Algorithm	Best time Complexity	Average time Complexity	Worst time Complexity
Linear search	O(1)	O(n)	O(n)
Sentinel Search	O(1)	O(n)	O(n)
Binary Search	O(1)	O(log n)	O(log n)
fibbonacci Search	O(1)	O(log n)	O(log n)