

Algorithms & Data Structures

Lab 11 (Searching Algorithms in Python)

EX1: (Sequential algorithm)

Function to implement sequential algorithm

```
def search(arr, s, x):  
    for i in range(0, s):  
        if arr[i] == x:  
            return i  
    return -1
```

Driver code to test above

```
s = int(input('Enter the size of the list: '))  
user_list = [0]*s  
for index in range(s):  
    user_list[index] = int(input('Enter a number: '))  
x = int(input('Enter the number you looking for: '))  
result = search(user_list, s, x)  
if result == -1:  
    print("Element is not present in array")  
else:  
    print("User list: ", user_list)  
    print("Element is present at index", result)
```

EX2: (Binary algorithm)

```
# Function to implement sequential algorithm
```

```
def search(arr, l, r, x):
    # Check base case
    if r >= l:
        mid = l + (r - l) // 2
        # If element is present at the middle itself
        if arr[mid] == x:
            return mid
        # If element is smaller than mid, then it
        # can only be present in left subarray
        elif arr[mid] > x:
            return search(arr, l, mid - 1, x)
        # Else the element can only be present
        # in right subarray
        else:
            return search(arr, mid + 1, r, x)

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

# Driver code to test above
s = int(input('Enter the size of the list: '))
user_list = [0]*s
for index in range(s):
    user_list[index] = int(input('Enter a number: '))
```

```
x = int(input('Enter the number you looking for: '))
# First, we need to sort the list
user_list.sort()
result = search(user_list, 0, s, x)
if result == -1:
    print("Element is not present in array")
else:
    print("User list: ", user_list)
    print("Element is present at index", result)
```