

DATA STRUCTURES LABORATORY MANUAL

– ICE 2144

III SEMESTER B. TECH

EXPT. 11 SORTING AND SEARCHING using Python

1. Sort a List in Ascending Order Without using an Extra Variable

#Python program to print a list

without using the sort() function & without an extra variable

```
l1=[76, 23, 45, 12, 54, 9]
```

```
print("Original List:", l1)
```

```
# sorting list using nested loops
```

```
for i in range(0, len(l1)):
```

```
    for j in range(i+1, len(l1)):
```

```
        if l1[i] >= l1[j]:
```

```
            l1[i], l1[j] = l1[j], l1[i]
```

```
# sorted list
```

```
print("Sorted List", l1)
```

2. Sort a List in Ascending Order by Using an extra variable

#Python program to print a list

without using the sort() function

using an extra variable

```
l1=[76,23,45,12,54,9]
```

```
print("Original List:", l1)
```

```
# sorting list using nested loops
```

```
for i in range(0, len(l1)):
```

DATA STRUCTURES LABORATORY MANUAL

– ICE 2144

III SEMESTER B. TECH

```
for j in range(i+1, len(l1)):
    if l1[i] >= l1[j]:
        # temporary variable
        temp = l1[i]
        l1[i] = l1[j]
        l1[j] = temp

# sorted list
print("Sorted List", l1)
```

3. Bubble Sort of a given List

```
def bubblesort(list):

    # Swap the elements to arrange in order
    for iter_num in range(len(list)-1,0,-1):
        for idx in range(iter_num):
            if list[idx]>list[idx+1]:
                temp = list[idx]
                list[idx] = list[idx+1]
                list[idx+1] = temp

list = [19,2,31,45,6,11,121,27]
bubblesort(list)
print(list)
```

DATA STRUCTURES LABORATORY MANUAL

– ICE 2144

III SEMESTER B. TECH

4. Merge Sort of a given List

```
def merge_sort(unsorted_list):
    if len(unsorted_list) <= 1:
        return unsorted_list
    # Find the middle point and divide it
    middle = len(unsorted_list) // 2
    left_list = unsorted_list[:middle]
    right_list = unsorted_list[middle:]

    left_list = merge_sort(left_list)
    right_list = merge_sort(right_list)
    return list(merge(left_list, right_list))

# Merge the sorted halves
def merge(left_half, right_half):
    res = []
    while len(left_half) != 0 and len(right_half) != 0:
        if left_half[0] < right_half[0]:
            res.append(left_half[0])
            left_half.remove(left_half[0])
        else:
            res.append(right_half[0])
            right_half.remove(right_half[0])
    if len(left_half) == 0:
        res = res + right_half
    else:
        res = res + left_half
```

DATA STRUCTURES LABORATORY MANUAL

– ICE 2144

III SEMESTER B. TECH

```
return res

unsorted_list = [64, 34, 25, 12, 22, 11, 90]

print(merge_sort(unsorted_list))
```

5. Linear Search of a given List

```
def linear_search(values, search_for):
    search_at = 0
    search_res = False
    # Match the value with each data element
    while search_at < len(values) and search_res is False:
        if values[search_at] == search_for:
            search_res = True
        else:
            search_at = search_at + 1
    return search_res

l = [64, 34, 25, 12, 22, 11, 90]
print(linear_search(l, 12))
print(linear_search(l, 91))
```

Exercise:

1. Write a program (in .py) to sort an array in descending order:
 - a. Using an extra variable
 - b. Without an extra variable
2. Write a program (in .py) to sort an array using HEAP SORT algorithm
3. Write a program (in .py) to perform binary search for an unsorted array.