Name: Deeptej Dhauskar | Roll no: 211105017

Lab session 11: Searching and Sorting Algorithms

Linear Search

```
#include <stdio.h>

int Linear_Search(int arr[], int n,
int item)
{
    int i = 0;
    while (i < n && arr[i] != item)
        i++;
    if (i < n)
        return i;
    return -1;
}

int main(int argc, char const *argv[])
{
    int n;
    printf("Enter number: ");
    scanf("%d", &n);
    int arr[n];
    for (int i = 0; i < n; i++)
    {
        printf("Enter number %d: ", i
+ 1);
        scanf("%d", &arr[i]);
    }
    printf("Enter item to be searched:
");
    int item, loc;
    scanf("%d", &item);
    if ((loc = Linear_Search(arr, n,
item)) != -1)
        printf("%d found at index
%d\n", item, loc);
    else
        printf("%d not present\n",
item);
    return 0;
}
```

Output:

```
PS
C:\Users\Deeptej\Desktop\Data-Structur
es-Lab\Algorithms> &
.\"Linear_Search.exe"
s> & .\"Linear_Search.exe"
Enter number: 5
Enter number 1: 12
Enter number 2: 45
Enter number 3: 87
Enter number 4: 23
Enter number 5: 90
Enter item to be searched: 23
23 found at index 3
```

Binary Search

```
#include <stdio.h>

int binarySearchNR(int arr[], int
item, int n, int low, int up)
{
    int mid;
    while (low <= up)
    {
        mid = (low + up) / 2;
        if (item < arr[mid])
            up = mid - 1;
        else if (item > arr[mid])
            low = mid + 1;
        else
            return mid;
    }
    return -1;
}

void Insertion_Sort(int arr[], int n)
{
    for (int i = 1; i < n; i++)
    {
        int j = i - 1, k = arr[i];
        while (j >= 0 && k < arr[j])
        {
            arr[j + 1] = arr[j];
            j--;
        }
        arr[j + 1] = k;
    }
}

int binarySearchR(int arr[], int item,
int low, int up)
{
    int mid;
    if (low > up)
        return -1;
    mid = (low + up) / 2;
    if (item < arr[mid])
        binarySearchR(arr, item, low,
mid - 1);
    else if (item > arr[mid])
        binarySearchR(arr, item, mid +
1, up);
    else
        return mid;
}

int main(int argc, char const *argv[])
{
    int n;
    printf("Enter number: ");
    scanf("%d", &n);
    int arr[n], loc;
    for (int i = 0; i < n; i++)
    {
        printf("Enter number %d: ", i
+ 1);
        scanf("%d", &arr[i]);
    }
    Insertion_Sort(arr, n);
    printf("After Sorting: ");
    for (int i = 0; i < n; i++)
        printf("%3d", arr[i]);
    printf("\nEnter item to be
searched: ");
    int item;
    scanf("%d", &item);
    if ((loc = binarySearchNR(arr,
item, n, 0, n - 1)) != -1)
        printf("%d found at index
%d\n", item, loc);
    else
        printf("%d not found\n",
item);
    printf("Enter item to be searched:
");
    scanf("%d", &item);
    if ((loc = binarySearchR(arr,
item, 0, n - 1)) != -1)
        printf("%d found at index
%d\n", item, loc);
    else
        printf("%d not found\n",
item);
    return 0;
}
```

Output:

```
PS
C:\Users\Deeptej\Desktop\Data-Structur
es-Lab\Algorithms> &
.\"Binary_Search.exe"
Enter number: 5
Enter number 1: 1
Enter number 2: 34
Enter number 3: 67
Enter number 4: 23
Enter number 5: 13
After Sorting:   1 13 23 34 67
Enter item to be searched: 23
23 found at index 2
Enter item to be searched: 55
55 not found
```

<u>Bubble Sort</u>

```c
#include <stdio.h>

void show(int arr[], int n)
{
    for (int i = 0; i < n; i++)
        printf("%d, ", arr[i]);
    printf("\n");
}

int main(int argc, char const *argv[])
{
    int n, temp;
    printf("Enter number: ");
    scanf("%d", &n);
    int arr[n];
    for (int i = 0; i < n; i++)
    {
        printf("Enter number %d: ", i
+ 1);
        scanf("%d", &arr[i]);
    }
    show(arr, n);
    for (int i = 0; i < n - 1; i++) //
no of passes
    {
        int swaps = 0;
        for (int j = 0; j < n - i - 1;
j++) // exclusing the last elements
cuz they sorted
        {
            if (arr[j] > arr[j + 1])
            {
                temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
                swaps++;
            }
        }
        if (!swaps)
            break;
    }
    printf("Sorted list: ");
    show(arr, n);
    return 0;
}
```

Output:

```
PS
C:\Users\Deeptej\Desktop\Data-Structur
es-Lab\Algorithms> &
.\"Bubble_Sort.exe"
Enter number: 10
Enter number 1: 49
Enter number 2: 75
Enter number 3: 47
Enter number 4: 10
Enter number 5: 13
Enter number 6: 25
Enter number 7: 49
Enter number 8: 99
Enter number 9: 15
Enter number 10: 44
49, 75, 47, 10, 13, 25, 49, 99, 15,
44,
Sorted list: 10, 13, 15, 25, 44, 47,
49, 49, 75, 99,
```

Selection Sort

```c
#include <stdio.h>

int isMin(int a, int b)
{
    return (a < b);
}

void swapnum(int *a, int *b)
{
    int temp = *a;
    *a = *b;
    *b = temp;
}

void show(int arr[], int n)
{
    for (int i = 0; i < n; i++)
        printf("%d, ", arr[i]);
    printf("\n");
}

int main(int argc, char const *argv[])
{
    int n;
    printf("Enter number of elements:
");
    scanf("%d", &n);
    int arr[n];
    for (int i = 0; i < n; i++)
    {
        printf("Enter element %d: ", i
+ 1);
        scanf("%d", &arr[i]);
    }
    printf("Before Sort: ");
    show(arr, n);
    int min;
    for (int i = 0; i < n; i++)
    {
        min = i; // holds index of min
element
        for (int j = i + 1; j < n;
j++)
        {
            if (isMin(arr[j],
arr[min]))
            {
                min = j;
            }
        }
        swapnum(&arr[i], &arr[min]);
    }
    printf("After Sort:  ");
    show(arr, n);
    return 0;
}
```

Output:

```
PS
C:\Users\Deeptej\Desktop\Data-Structur
es-Lab\Algorithms> &
.\"selection_sort.exe"
Enter number of elements: 10
Enter element 1: 49
Enter element 2: 75
Enter element 3: 47
Enter element 4: 10
Enter element 5: 13
Enter element 6: 25
Enter element 7: 49
Enter element 8: 99
Enter element 9: 15
Enter element 10: 44
Before Sort: 49, 75, 47, 10, 13, 25,
49, 99, 15, 44,
After Sort:  10, 13, 15, 25, 44, 47,
49, 49, 75, 99,
```

Insertion Sort

```
#include <stdio.h>

int main(int argc, char const *argv[])
{
    int n;
    printf("Enter number of elements:
");
    scanf("%d", &n);
    int arr[n];
    printf("Insert elements\n");
    for (int i = 0; i < n; i++)
        scanf("%d", &arr[i]);
    for (int i = 1; i < n; i++)
    {
        int j = i - 1, k = arr[i];
        while (j >= 0 && k < arr[j])
        {
            arr[j + 1] = arr[j];
            j--;
        }
        arr[j + 1] = k;
    }
    printf("Sorted array is: ");
    for (int i = 0; i < n; i++)
```

```
        printf("%3d", arr[i]);

    return 0;
}
```

Output:

```
PS
C:\Users\Deeptej\Desktop\Data-Structur
es-Lab\Algorithms> &
.\"Insertion_Sort.exe"
Enter number of elements: 10
Insert elements
49
75
47
10
13
25
49
99
15
44
Sorted array is:  10 13 15 25 44 47 49
49 75 99
```

Merge Sort

```
#include <stdio.h>
void merge(int arr[], int left, int
mid, int right)
{
    int i, j, k;
    int n1 = mid - left + 1;
    int n2 = right - mid; // right
minus mid because mid actually lies
before the right part
    int L[n1], R[n2];
    // splitting the parent array
    for (i = 0; i < n1; i++)
        L[i] = arr[left + i];
    for (j = 0; j < n2; j++)
        R[j] = arr[mid + 1 + j]; //
for right half left is mid+1
    i = 0;
    j = 0;
    k = left;
    while (i < n1 && j < n2)
    {
        if (L[i] <= R[j])
        {
            arr[k] = L[i];
            i++;
        }
        else
        {
            arr[k] = R[j];
            j++;
        }
        k++;
    }
    while (i < n1)
    {
        arr[k] = L[i];
        i++;
        k++;
    }
    while (j < n2)
    {
        arr[k] = R[j];
        j++;
        k++;
    }
}
void mergesort(int arr[], int left,
int right)
{
    if (left < right)
    {
        int mid = left + (right -
left) / 2;
        mergesort(arr, left, mid);
        mergesort(arr, mid + 1,
right);

        merge(arr, left, mid, right);
    }
}
void printArray(int arr[], int n)
{
    for (int i = 0; i < n; i++)
        printf("%d, ", arr[i]);
    printf("\n");
}
int main(int argc, char const *argv[])
{
    // printf("Enter number of
elements: ");
    // scanf("%d", &n);
    int arr[] = {28, 66, 16, 76, 71,
86, 94, 97, 56, 95};
    int n = sizeof(arr) /
sizeof(arr[0]);
    // printf("Insert elements\n");
    // for (int i = 0; i < n; i++)
    //     scanf("%d", &arr[i]);
    printArray(arr, n);
    printf("Sorted array: ");
    mergesort(arr, 0, n - 1);
    printArray(arr, n);
    return 0;
}
```

Output:

```
PS
C:\Users\Deeptej\Desktop\Data-Structur
es-Lab\Algorithms> &
.\"merge_sort.exe"
28, 66, 16, 76, 71, 86, 94, 97, 56,
95,
Sorted array: 16, 28, 56, 66, 71, 76,
86, 94, 95, 97,
```