Design and Analysis of Algorithms

Week-2 Assignment

1. Bubble Sort

PROGRAM:

```
#include <stdio.h>
int main() {
int n;
printf("ENTER NO.OF.ELEMENTS:");
scanf("%d",&n);
int arr[n];
printf("ENTER THE VALUES:");
for(int i=0;i < n;i++){
scanf("%d",&arr[i]);
}
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
    printf("Sorted array: ");
    for (int i = 0; i < n; i++) {
        printf("\t%d\t,",arr[i]);
    }
    return 0;
}
```

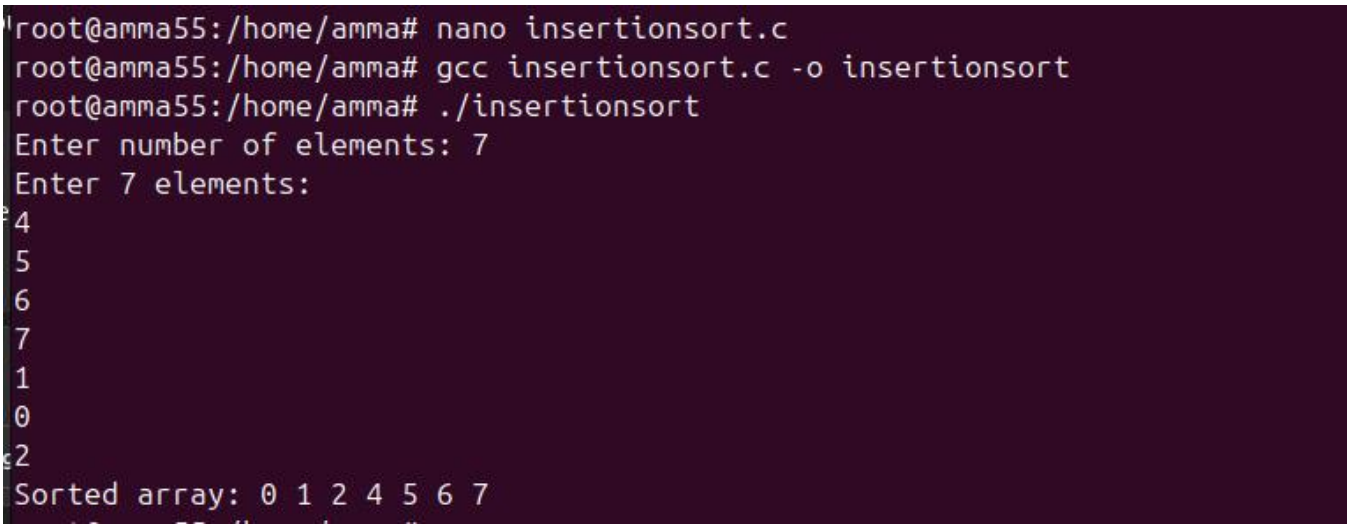
```
root@amma55:/home/amma# nano bubblesort.c
root@amma55:/home/amma# gcc bubblesort.c -o bubblesort
root@amma55:/home/amma# ./bubblesort
n:6
arr[0]:5
arr[1]:6
arr[2]:7
arr[3]:8
arr[4]:2
arr[5]:1
Sorted array: 1 2 5 6 7 8
```

2. Insertion Sort

PROGRAM:

```
#include <stdio.h>
int main() {
    int n;
    printf("ENTER NO.OF ELEMENTS");
    scanf("%d",&n);
    int arr[n];
    printf("ENTER THE ELEMENTS:");
    for(int i=0; i <n;i++){
    scanf("%d",&arr[i]);
    }
    for (int i = 1; i < n; i++) {
        int key = arr[i];
        int j = i - 1;
        while (j >= 0 && arr[j] > key) {
            arr[j + 1] = arr[j];
```

```
            j--;
        }
        arr[j + 1] = key;
    }
    printf("Sorted array: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    return 0;
}
```



3. Selection Sort

PROGRAM:

```
#include <stdio.h>
int main() {
    int n;
    printf("ENTER NO. OF ELEMENTS: ");
    scanf("%d", &n);
    int arr[n];
    printf("ENTER THE VALUES: ");
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    for (int i = 0; i < n - 1; i++) {
        int minIndex = i;
        for (int j = i + 1; j < n; j++) {
            if (arr[j] < arr[minIndex]) {
                minIndex = j;
            }
        }
        int temp = arr[i];
        arr[i] = arr[minIndex];
        arr[minIndex] = temp;
```

```
    }
    printf("Sorted array: ");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    return 0;
}
```

```
root@amma55:/home/amma# nano selectionsort.c
root@amma55:/home/amma# gcc selectionsort.c -o selectionsort
root@amma55:/home/amma# ./selectionsort
Enter number of elements: 5
Enter 5 elements:
5
6
7
3
2
Sorted array: 2 3 5 6 7
```

4. Bucket Sort

PROGRAM:

```
#include <stdio.h>
```

```
int main() {
    int n;
    printf("ENTER NO. OF ELEMENTS: ");
    scanf("%d", &n);
    int arr[n];
    printf("ENTER THE VALUES (0 to 100): ");
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    int bucket[101] = {0};
    for (int i = 0; i < n; i++) {
        bucket[arr[i]]++;
    }
    printf("Sorted array: ");
    for (int i = 0; i <= 100; i++) {
        while (bucket[i] > 0) {
            printf("%d ", i);
            bucket[i]--;
        }
    }
    return 0;
}
```

```
root@amma11:/home/amma# nano bucketsort.c
root@amma11:/home/amma# gcc bucketsort.c -o bucketsort
root@amma11:/home/amma# ./bucketsort
ENTER NO. OF ELEMENTS: 6
ENTER THE VALUES (0 to 100): 43
21
76
3
98
34
Sorted array: 3 21 34 43 76 98
```

5. Heap Sort

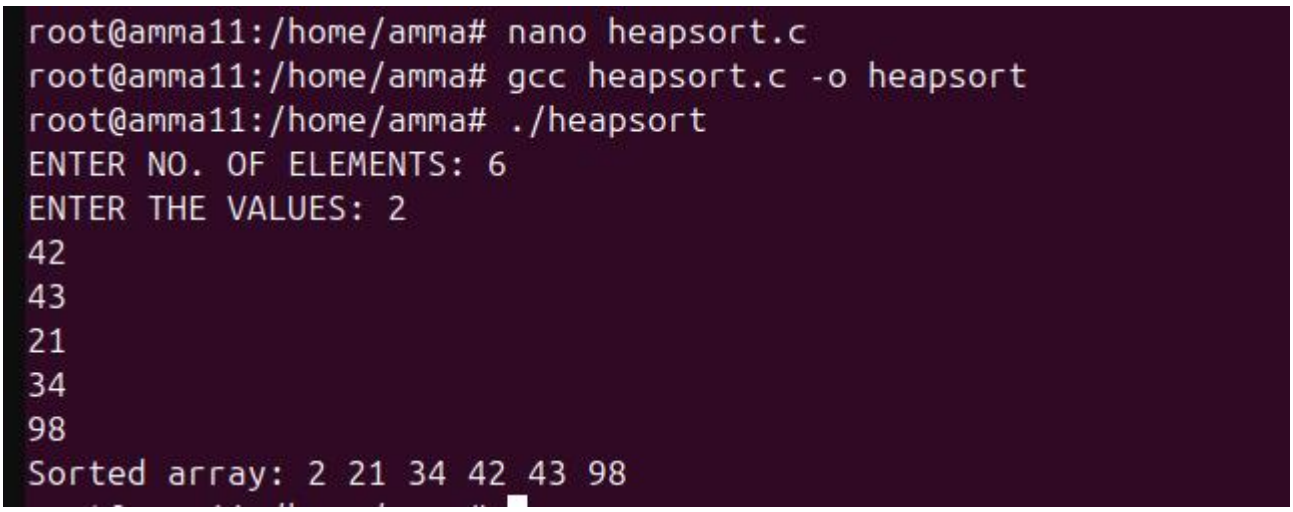
PROGRAM:

```
#include<stdio.h>
int main() {
    int n;
    printf("ENTER NO. OF ELEMENTS: ");
    scanf("%d", &n);
    int arr[n];
    printf("ENTER THE VALUES: ");
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }
    for (int i = 1; i < n; i++) {
        int child = i;
        while (child > 0) {
            int parent = (child - 1) / 2;
```

```
        if (arr[parent] < arr[child]) {
            int temp = arr[parent];
            arr[parent] = arr[child];
            arr[child] = temp;
            child = parent;
        } else {
            break;
        }
    }
}
for (int i = n - 1; i > 0; i--) {
    int temp = arr[0];
    arr[0] = arr[i];
    arr[i] = temp;
    int parent = 0;
    while (1) {
        int left = 2 * parent + 1;
        int right = 2 * parent + 2;
        int largest = parent;
        if (left < i && arr[left] > arr[largest])
            largest = left;
        if (right < i && arr[right] > arr[largest])
```

```
            largest = right;
        if (largest != parent) {
            int temp2 = arr[parent];
            arr[parent] = arr[largest];
            arr[largest] = temp2;
            parent = largest;
        } else {
            break;
        }
    }
  }
  printf("Sorted array: ");
  for (int i = 0; i < n; i++) {
    printf("%d ", arr[i]);
  }
    printf("\n");

  return 0;
}
```



DONIRAJ SALINDRA
CH.SC.U4CSE24245