WEEK-2

DESIGN AND ANALYSIS ALGORITHM

NAME: KANTAMRAJU SREEJA

ROLL NO: CH.SC.U4CSE24255

1) BUBBLE SORT:

CODE:

```c
#include <stdio.h>
int main() {
int n, i, j, temp;
int arr[100];
printf("Enter number of elements: ");
scanf("%d", &n);
printf("Enter %d elements:\n", n);
for (i = 0; i < n; i++) {
scanf("%d", &arr[i]);
}
for (i = 0; i < n - 1; i++) {
for (j = 0; j < n - 1 - i; j++) {
if (arr[j] > arr[j + 1]) {
temp = arr[j];
arr[j] = arr[j + 1];
arr[j + 1] = temp;
}
}
}
printf("Sorted array:\n");
for (i = 0; i < n; i++) {
printf("%d ", arr[i]);
}
printf("\n");
return 0;
}
```

OUTPUT:

```
kantamraju@kantamraju-VirtualBox:~$ gcc bubblesort.c -o bubblesort
kantamraju@kantamraju-VirtualBox:~$ ./bubblesort
Enter number of elements: 4
Enter 4 elements:
4 2 5 8
Sorted array:
2 4 5 8
```

2) INSERTION SORT:

CODE:

```c
#include <stdio.h>
int main() {
int n, i, j, key;
int arr[100];
printf("Enter number of elements: ");
scanf("%d", &n);
printf("Enter %d elements:\n", n);
for (i = 0; i < n; i++) {
scanf("%d", &arr[i]);
}
for (i = 1; i < n; i++) {
key = arr[i];
j = i - 1;
while (j >= 0 && arr[j] > key) {
arr[j + 1] = arr[j];
j--;
}
arr[j + 1] = key;
}
printf("Sorted array:\n");
for (i = 0; i < n; i++) {
printf("%d ", arr[i]);
}
printf("\n");
return 0;
}
```

OUTPUT:

```
kantamraju@kantamraju-VirtualBox:~$ gcc insertionsort.c -o insertionsort
kantamraju@kantamraju-VirtualBox:~$ ./insertionsort
Enter number of elements: 5
Enter 5 elements:
8 6 9 3 1
Sorted array:
1 3 6 8 9
```

3)SELECTION SORT:

CODE:

```c
#include <stdio.h>
int main() {
int n, i, j, minIndex, temp;
int arr[100];
printf("Enter number of elements: ");
scanf("%d", &n);
printf("Enter %d elements:\n", n);
for (i = 0; i < n; i++) {
scanf("%d", &arr[i]);
}
for (i = 0; i < n - 1; i++) {
minIndex = i;
for (j = i + 1; j < n; j++) {
if (arr[j] < arr[minIndex]) {
minIndex = j;
}
}
if (minIndex != i) {
temp = arr[i];
arr[i] = arr[minIndex];
arr[minIndex] = temp;
}
}
printf("Sorted array:\n");
for (i = 0; i < n; i++) {
printf("%d ", arr[i]);
```

```c
for (j = i + 1; j < n; j++) {
if (arr[j] < arr[minIndex]) {
minIndex = j;
}
}
if (minIndex != i) {
temp = arr[i];
arr[i] = arr[minIndex];
arr[minIndex] = temp;
}
}
printf("Sorted array:\n");
for (i = 0; i < n; i++) {
printf("%d ", arr[i]);
}
printf("\n");
return 0;
}
```

OUTPUT:

```
kantamraju@kantamraju-VirtualBox:~$ gcc selectionsort.c -o selectionsort
kantamraju@kantamraju-VirtualBox:~$ ./selectionsort
Enter number of elements: 3
Enter 3 elements:
8 2 6
Sorted array:
2 6 8
```

4) BUCKETSORT:

CODE:

```c
#include <stdio.h>
void bucketSort(int a[], int n) {
int b[101]={0};
for(int i=0;i<n;i++) b[a[i]]++;
int k=0;
for(int i=0;i<101;i++)
while(b[i]--) a[k++]=i;
}
int main() {
int n;
printf("Enter size: ");
scanf("%d",&n);
int a[n];
printf("Enter elements (0-100): ");
for(int i=0;i<n;i++){
scanf("%d",&a[i]);
}
printf("Before sorting: ");
for(int i=0;i<n;i++){
printf("%d ",a[i]);
}
```

```c
while(b[i]--) a[k++]=i;
}
int main() {
int n;
printf("Enter size: ");
scanf("%d",&n);
int a[n];
printf("Enter elements (0-100): ");
for(int i=0;i<n;i++){
scanf("%d",&a[i]);
}
printf("Before sorting: ");
for(int i=0;i<n;i++){
printf("%d ",a[i]);
}
bucketSort(a,n);
printf("\nAfter Bucket Sort: ");
for(int i=0;i<n;i++) {
printf("%d ",a[i]);
}
printf("\n");
return 0;
}
```

OUTPUT:

```
kantamraju@kantamraju-VirtualBox:~$ gcc bubblesort.c -o bucketsort
kantamraju@kantamraju-VirtualBox:~$ ./bubblesort
Enter number of elements: 5
Enter 5 elements:
2 8 5 9 3
Sorted array:
2 3 5 8 9
```

5)HEAP SORT:

CODE:

```c
#include <stdio.h>
void heapify(int arr[], int n, int i) {
int largest = i;
int left = 2 * i + 1;
int right = 2 * i + 2;
int temp;
if (left < n && arr[left] > arr[largest])
largest = left;
if (right < n && arr[right] > arr[largest])
largest = right;
if (largest != i) {
temp = arr[i];
arr[i] = arr[largest];
arr[largest] = temp;
heapify(arr, n, largest);
}
}
void heapSort(int arr[], int n) {
int i, temp;
for (i = n / 2 - 1; i >= 0; i--)
heapify(arr, n, i);
for (i = n - 1; i > 0; i--) {
temp = arr[0];
arr[0] = arr[i];
arr[i] = temp;
heapify(arr, i, 0);
```

```
for (i = n - 1; i > 0; i--) {
temp = arr[0];
arr[0] = arr[i];
arr[i] = temp;
heapify(arr, i, 0);
}
}
int main() {
int n, i;
int arr[100];
printf("Enter number of elements: ");
scanf("%d", &n);
printf("Enter %d elements:\n", n);
for (i = 0; i < n; i++) {
scanf("%d", &arr[i]);
}
heapSort(arr, n);
printf("Sorted array:\n");
for (i = 0; i < n; i++) {
printf("%d ", arr[i]);
}
printf("\n");
return 0;
}
```

OUTPUT:

```
kantamraju@kantamraju-VirtualBox:~$ gcc heapsort.c -o heapsort
kantamraju@kantamraju-VirtualBox:~$ ./heapsort
Enter number of elements: 4
Enter 4 elements:
3 8 1 6
Sorted array:
1 3 6 8
```