NAME: SOORAJ SHUKLA
ROLL NUMBER: 54
SEMESTER: 5TH

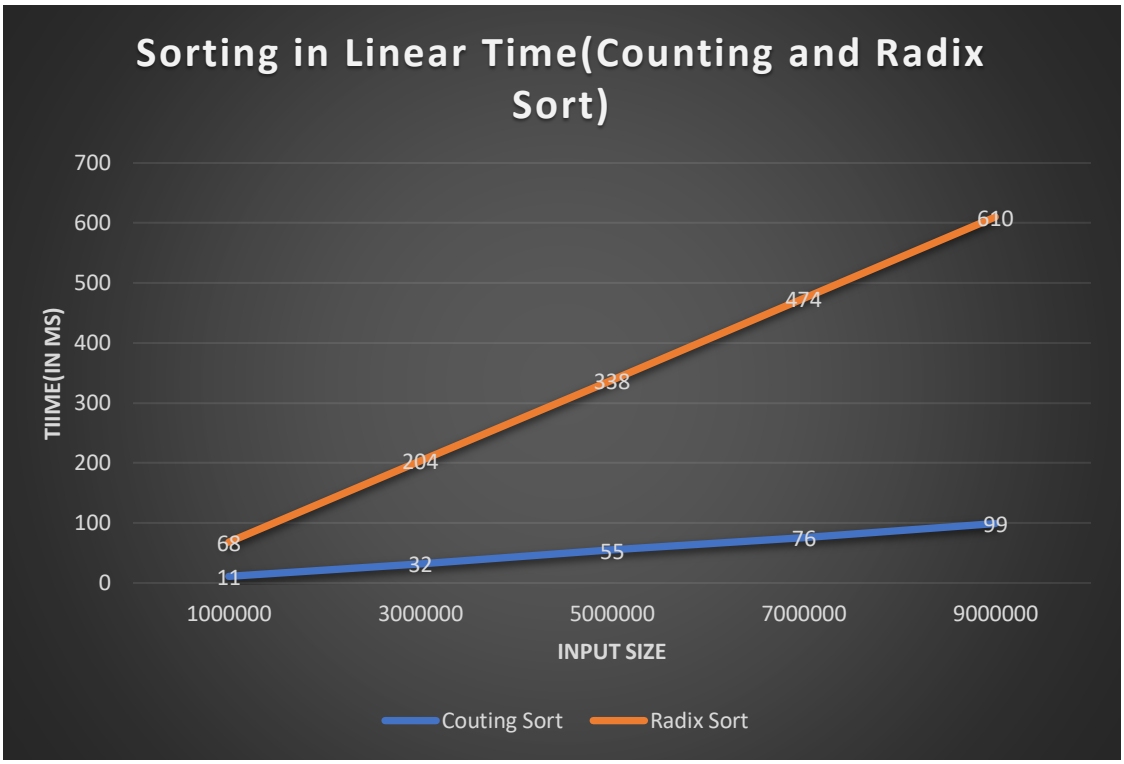
```
int getMax(int* arr, int n){
    int max = -1;
    for(int i = 0; i < n; i++)
        if(max < arr[i])
            max = arr[i];
    return max;
}
void CountingSort(int*arr, int n){
    //finding range
    int max = getMax(arr,n);
    max++;
    int* posArr = (int*)malloc(max*sizeof(int));
    for(int i = 0; i < max; i++)
        posArr[i] = 0;
    for(int i = 0; i < n; i++)
        posArr[arr[i]]++;
    for(int i = 1; i < max; i++)
        posArr[i] += posArr[i-1];
    int* sortedArr = (int*)malloc(n*sizeof(int));
    for(int i = n-1; i >= 0; i--)
        sortedArr[--posArr[arr[i]]] = arr[i];
    for(int i = 0; i < n; i++)
        arr[i] = sortedArr[i];
}
void useCountingSort(int* arr, int n, int e){
    int* r = (int*)malloc(n*sizeof(int));
    int i, count[10] = {0,0,0,0,0,0,0,0,0,0};

    for (i = 0; i < n; i++)
        count[(arr[i]/e)%10]++;

    for (i = 1; i < 10; i++)
        count[i] += count[i - 1];

    for (i = n - 1; i >= 0; i--)
        r[--count[(arr[i]/e)%10]] = arr[i];
    for (i = 0; i < n; i++)
        arr[i] = r[i];
}
void radixSort(int* arr, int n){
    int m = getMax(arr, n);
    for (int e = 1; m/e > 0; e *= 10)
        useCountingSort(arr, n, e);
}
```

GRAPH:


Sorting in Linear Time(Counting and Radix Sort)
TIME(IN MS)
700
600
500
400
300
200
100
0
68
204
338
474
610
11
32
55
76
99
1000000
3000000
5000000
7000000
9000000
INPUT SIZE
Couting Sort
Radix Sort