# KIET Group of Institutions, Ghaziabad

## Department of Computer Applications

**(An ISO – 9001: 2015 Certified & 'A' Grade accredited Institution by NAAC)**

## Design and Analysis of Algorithm

## RCA 352: Session 2020-21

## DAA Lab

### Experiment-No.

**Objective**: Implement the **Radix sort** algorithm to sort the given list of N numbers and plot graph

| Scheduled Date: | Compiled Date: | Submitted Date: |
|---|---|---|
| 15/10/2020 | 16/10/2020 | 18/10/2020 |

**Algorithm:**

```
void countSort(Input Array arr, Array Size: n, Passing Digit exp)
        Arr: Input Array
        n:Length of an Array arr
        exp: Passing digits
        output,cout:External array
        i,count: External variable

1.int output[100] // output array
2.int i, cout[10] = { 0 }
3. count++
4.for (i = 0; i < n; i++)
5.      count++
6.      cout[(arr[i] / exp) % 10]++;
7.      count++;
8.  end for;
9.  count++;
10.  for (i = 1; i < 10; i++)
11.     count++;
12.     cout[i] += cout[i - 1];
13.     count++;
14.  end of for;
15.  count++
for (i = n - 1; i >= 0; i--)


      count++;
      output[cout[(arr[i] / exp) % 10] - 1] = arr[i];
      count++;
      cout[(arr[i] / exp) % 10]--;
      count++;
end of for;
 count++
for (i = 0; i < n; i++)
      count++;
      arr[i] = output[i];
      count++;
end of for;
end of countSort;
```

```
int getMax(Input Array arr,Array Size n)
mx,i,count: External Variable
int mx = arr[0];
int i;
count++;
for (i = 1; i < n; i++)
count++;
      if (arr[i] > mx)
          count++;
          mx = arr[i];
          count++;
      end if
```

```
11.         count++;
12.    end   for loop
13     return mx;
14.end getMax
```

```
void radixsort(Input Array  arr,Array Size n)
m:external Variable
1.     int m = getMax(arr, n)
2.             int exp
3.    count++
4.    for (exp = 1; m / exp > 0; exp *= 10)
5.          count++
6.          countSort(arr, n, exp)
7.          count++
8.    end for loop
9.end radixsort
```

**Program:**

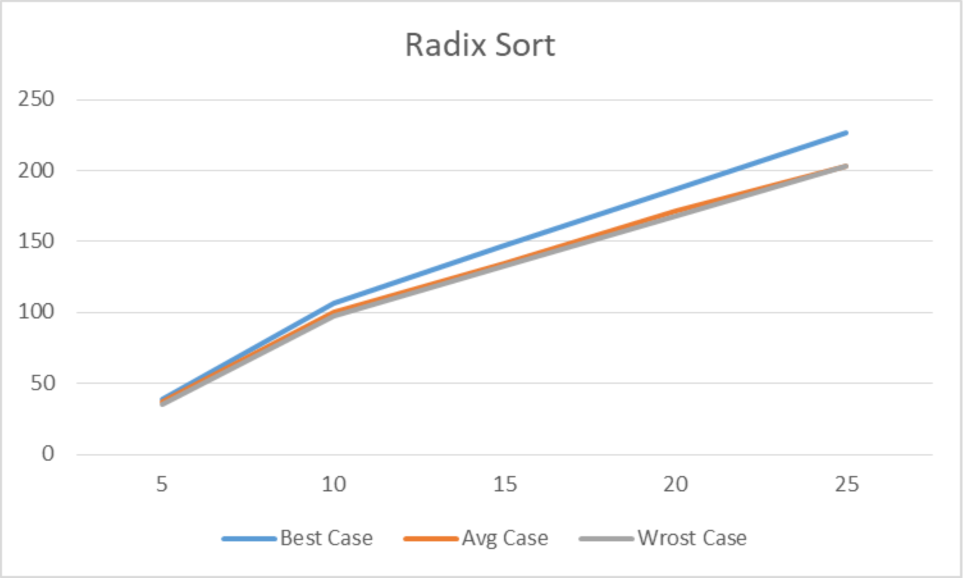
```
#include <stdio.h>
#include <conio.h>
#include <process.h>
#include <alloc.h>
int count=0;

void main()
{
   void radixsort(int a[],int);
   void print(int a[],int);
   int arr[100];
   int n,i;
   clrscr();
   printf("enter the sizee of array=");
   scanf("%d",&n);
   printf("enter the number=");
   for(i=0;i mx)
            {
               count++;
               mx = arr[i];
               count++;
            }
            count++;
   }
   return mx;
}

void countSort(int arr[], int n, int exp)
{
   int output[100]; // output array
   int i, cout[10] = { 0 };


   count++;
   for (i = 0; i < n; i++)
   {
            count++;
            cout[(arr[i] / exp) % 10]++;
            count++;
   }
   count++;
   for (i = 1; i < 10; i++)
   {
            count++;
            cout[i] += cout[i - 1];
            count++;
   }


   count++;
   for (i = n - 1; i >= 0; i--)
   {
            count++;
            output[cout[(arr[i] / exp) % 10] - 1] = arr[i];
            count++;
            cout[(arr[i] / exp) % 10]--;
            count++;
   }
    count++;
   for (i = 0; i < n; i++)
   {
            count++;

            arr[i] = output[i];
            count++;
   }
}
```

```
void radixsort(int arr[], int n)
 {
   int m = getMax(arr, n);

   int exp;
   count++;
   for (exp = 1; m / exp > 0; exp *= 10)
   {
            count++;
            countSort(arr, n, exp);
            count++;
   }
}

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

}
```

**Output**

| Input Size | Best Case | Average Case | Worst Case |
|---|---|---|---|
| 5 | 39 | 37 | 35 |
| 10 | 107 | 100 | 98 |
| 15 | 147 | 135 | 133 |
| 20 | 187 | 172 | 168 |
| 25 | 227 | 203 | 203 |

**Graph:**



**Conclusion:**

| Case | Running Time : Growth of Function mathematically | Running Time : Growth of Function after observing graph |
|---|---|---|
| Best Case | O(n) | O(n) |
| Average Case | O(n) | O(n) |
| Worst Case | O(n) | O(n) |