**Experiment-No.5**

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

|  |  |  |
| --- | --- | --- |
| **Scheduled Date:** | **Compiled Date:** | **Submitted Date:** |
| 04-09-2020 | 4-9-2020 | 8-9-2020 |

**Algorithm:**

**partition(a[], p, r)**

pivot = a[r]

i = lo // place for swapping

for j := p to r – 1 do

if a[j] <= pivot then

swap a[i] with a[j]

i = i + 1

swap a[i] with a[r]

return i

**partition\_r(a[], p, r)**

r = Random Number from p to r

Swap a[r] and a[r]

return partition(a, p, r)

**quicksort(a[], p, r)**

if( p< r)

q = partition\_r(a, p, r)

quicksort(a, p, q-1)

quicksort(a, q+1, r)

**Program:**

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

#include<conio.h>

int count=0;

int main()

{

void get\_data(int [],int);

void put\_data(int [],int);

int myrandom(int ,int);

int swap(int\*,int\*);

int partition(int[],int,int);

void Quick\_sort(int [],int,int);

int a[100];

int n;

clrscr();

printf("Enter the size of an array:\n");

scanf("%d",&n);

printf("Enter the array elements:\n");

get\_data(a,n);

printf("Array before sorting:\n");

put\_data(a,n);

Quick\_sort(a,0,n-1);

printf("\nArray after sorting:\n");

put\_data(a,n);

printf("\n\nFor n=%d no of counts are=%d",n,count);

getch();

}

int myrandom(int lower,int upper)

{

int num;

srand(time(0));

num=(rand()%(upper -lower +1))+lower;

return num;

}

int partition(int a[],int p,int r)

{

int x,i,j,y,temp;

y=myrandom(p,r-1);

temp=a[y];

a[y]=a[r];

a[r]=temp;

x=a[r];

count++;

i=p-1;

count++;

for(j=p;j<=r-1;j++)

{

count++;

if(a[j]<x)

{

count++;

i=i+1;

count++;

swap(&a[i],&a[j]);

}

}

swap(&a[i+1],&a[r]);

count++;

return (i+1);

}

void Quick\_sort(int a[],int p,int r)

{

int q;

count++;

if(p<r)

{

count++;

q=partition(a,p,r);

count++;

Quick\_sort(a,p,q-1);

count++;

Quick\_sort(a,q+1,r);

count++;

}

}

void get\_data(int arr[],int n)

{

int i;

for (i = 0; i < n; i++)

{

scanf("%d",&arr[i]);

}

}

void put\_data(int a[],int n)

{

int i;

for(i=0 ; i < n; i++)

{

printf("%d ",a[i]);

}

}

int swap(int \*a,int \*b)

{

int temp;

temp=\*a;

\*a=\*b;

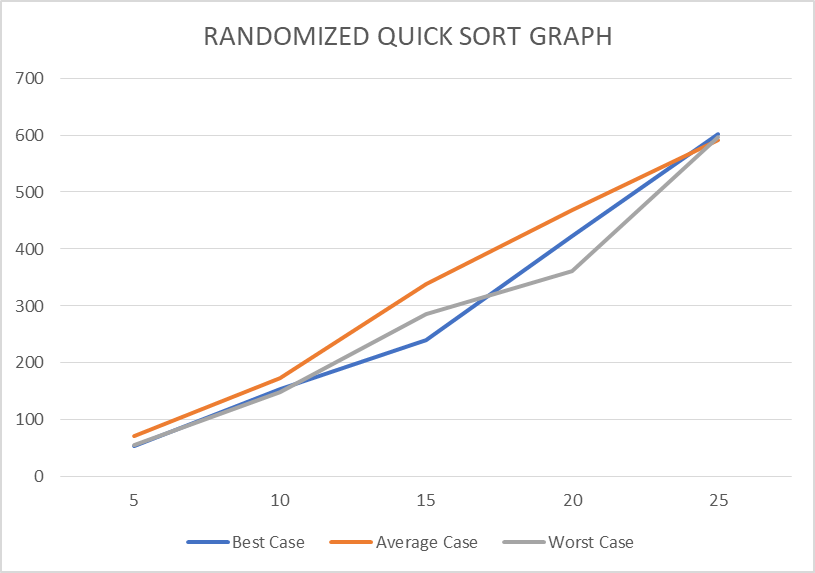
\*b=temp;

}

**Output:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Input Size** | **Best Case** | **Average Case** | **Worst Case** |
| 5 | 54 | 70 | 55 |
| 10 | 153 | 173 | 149 |
| 15 | 239 | 338 | 286 |
| 20 | 423 | 468 | 361 |
| 25 | 602 | 591 | 569 |

**Graph:**

****

**Conclusion:**

|  |  |  |
| --- | --- | --- |
| **Case** | **Running Time : Growth of Function mathematically** | **Running Time : Growth of Function after observing graph** |
| **Best Case** | O(nlogn) | O()nlogn |
| **Average Case** | O(nlogn) | O(nlogn) |
| **Worst Case** | O(n^2) | O(n^2) |