Experiment-No.4

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

|  |  |  |
| --- | --- | --- |
| Scheduled Date: | Compiled Date: | Submitted Date: |
| 26-8-2020 | 26-8-2020 | 29-8-2020 |

**Algorithm:**

Initial call is QUICKSORT (A, 1, n)

QUICKSORT (A, p, r)

1. if p < r

2. then q ←PARTITION(A, p, r )

3. QUICKSORT (A, p, q − 1)

4. QUICKSORT (A, q + 1, r)

Partitioning

Partition subarray A [p . . . r] by the following procedure:

PARTITION (A, p, r)

1. x ← A[r ]

2. i ←p –1

3. for j ←p to r –1

4. if A[ j ] ≤ x

5. i ←i + 1

6. exchange A[i ] ↔A[ j ]

7. exchange A[i + 1] ↔ A[r ]

8. return i + 1

Program file Quick\_sort.cpp:

#include<stdio.h>

#include<stdlib.h>

int count=0;

int main()

{

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

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

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

int a[100];

int n;

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);

return 0;

}

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

{

count++;

int x,i,j;

count++;

x=a[r];

count++;

i=p-1;

count++;

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

{

count++;

count++;

if(a[j]<x)

{

count++;

i=i+1;

count++;

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

count++;

}

}

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]);

}

}

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

{

int temp;

temp=\*a;

\*a=\*b;

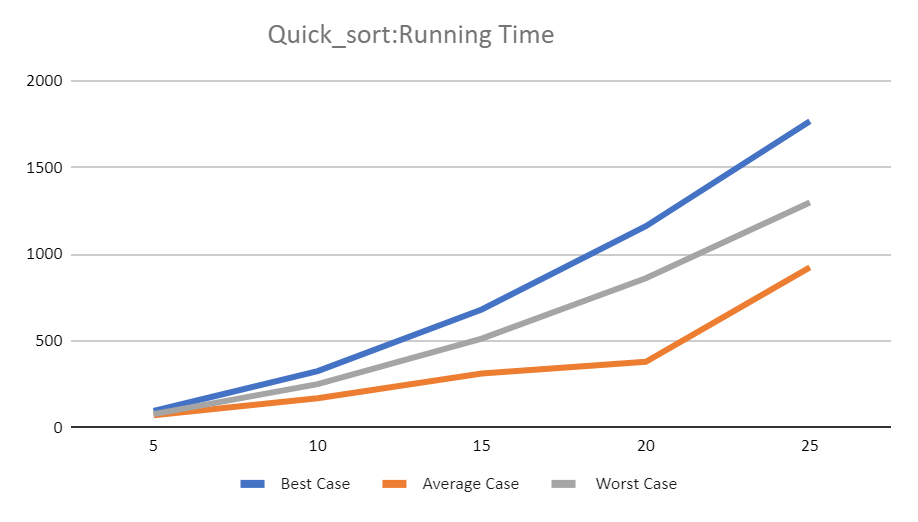
\*b=temp;

}

**Output:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Input Size** | **Best Case** | **Average Case** | **Worst Case** |
| 5 | 94 | 70 | 78 |
| 10 | 322 | 168 | 251 |
| 15 | 678 | 310 | 515 |
| 20 | 1159 | 376 | 861 |
| 25 | 1764 | 921 | 1298 |

**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(n2) | O(n2) |