DAA WEEK6 SUBMISSION

Name: Davasam Karthikeya   Section: AIMLB   Reg No: 230962326

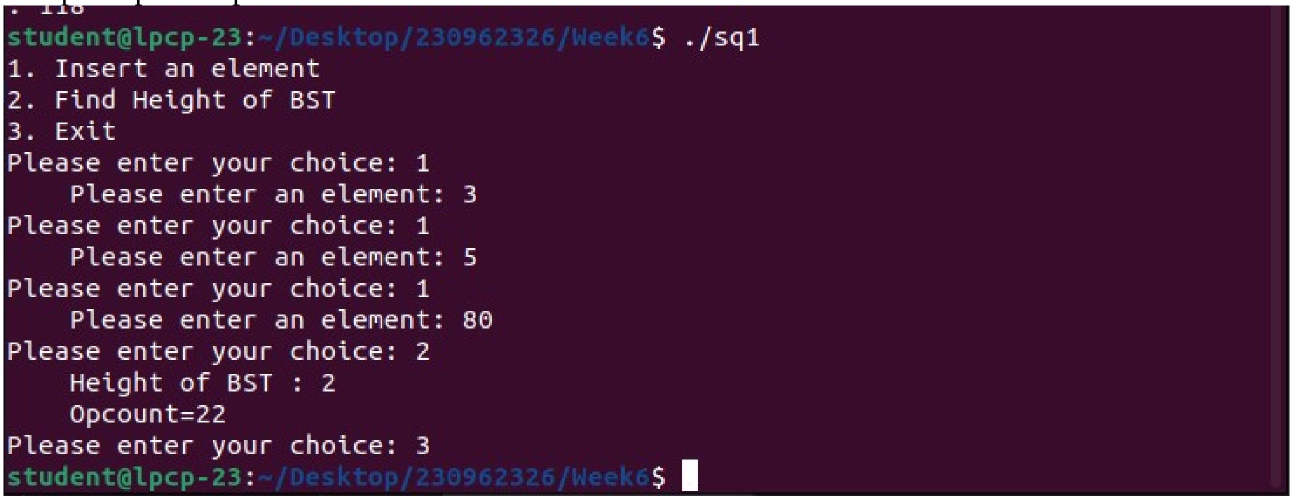
Date: 22/02/2025

1) Write a program to determine the height of a binary search tree and analyze its time efficiency.

Code:

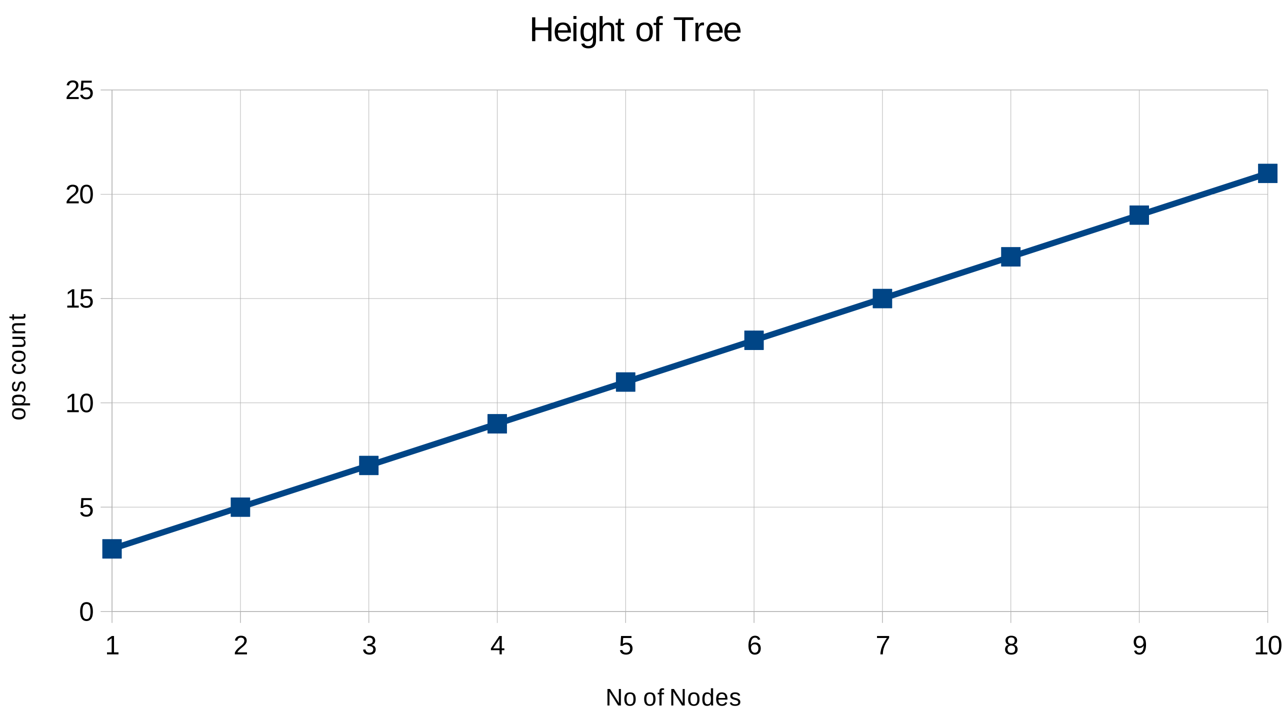
```c
#include<stdio.h>
#include<stdlib.h>
#define MAX(a,b) ((a) > (b) ? a : b)
int opcount=0;
struct node{
    int val;
    struct node *left, *right;
};

typedef struct node *NODE;
NODE root=NULL;
NODE insert(int ele, NODE node){
    NODE temp;
    if(node == NULL){
        temp= (NODE)malloc(sizeof(struct node));
        temp->val=ele;
        temp->left = temp->right=NULL;
        if(root == NULL)root=temp;
        return temp;
    }else if(ele < node->val){
        node->left = insert(ele, node->left);
        return node;
    }else if(ele > node->val){
        node->right = insert(ele,node->right);
        return node;
    }else {
        printf("Duplicate element found while inserting. Insertion failed\n");
        return NULL;
    }
}
int height(NODE node){
    opcount++;
    if(node==NULL)return -1;
    else return MAX(height(node->left), height(node->right))+1;
}
void main() {
    int choice,ele;
    printf("1. Insert an element\n");
    printf("2. Find Height of BST\n");
    printf("3. Exit\n");
    do{
        printf("Please enter your choice: ");
        scanf("%d",&choice);
        switch(choice) {
            case 1:
                printf("    Please enter an element: ");
                scanf("%d", &ele);
                insert(ele,root);
        break;
            case 2:
                opcount = 0;
                printf("    Height of BST : %d\n",height(root));
                printf("    Opcount=%d \n",opcount);
        break;
            case 3:
                break;
            default: printf("Invalid choice. Please enter valid choice\n");
        break;
        }
    }while(choice != 3);
}
```

Sample Input/Output:



```
student@lpcp-23:~/Desktop/230962326/Week6$ ./sq1
1. Insert an element
2. Find Height of BST
3. Exit
Please enter your choice: 1
    Please enter an element: 3
Please enter your choice: 1
    Please enter an element: 5
Please enter your choice: 1
    Please enter an element: 80
Please enter your choice: 2
    Height of BST : 2
    Opcount=22
Please enter your choice: 3
student@lpcp-23:~/Desktop/230962326/Week6$
```

Graph:



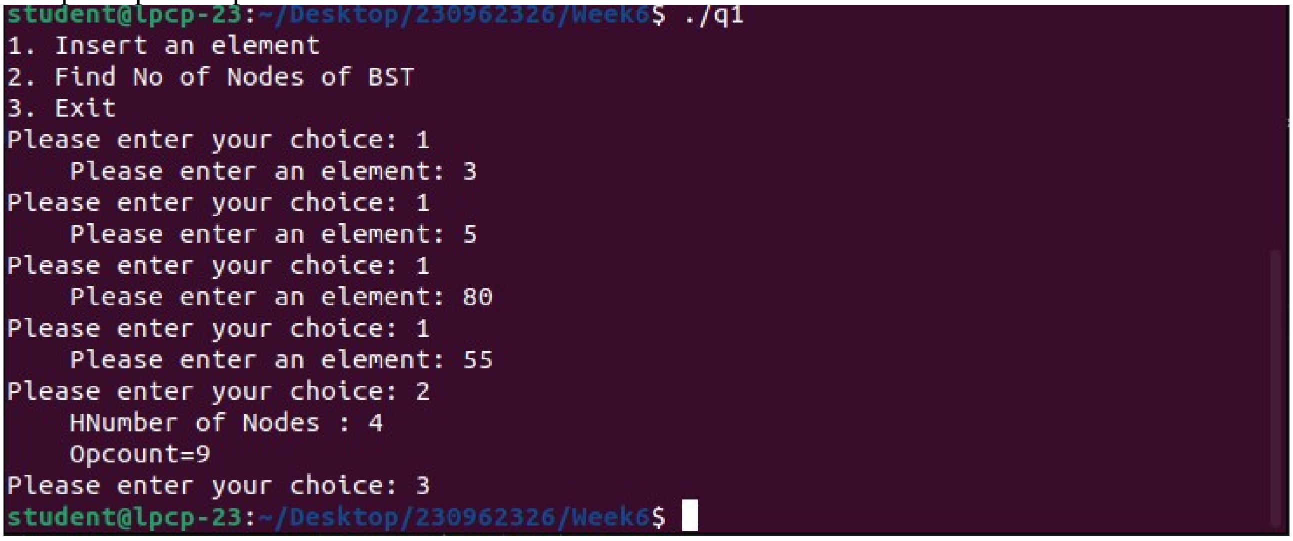
2) Find total number of nodes in a binary tree and analyze its efficiency. Obtain the experimental result of order of growth and plot the result.

Code:

```
#include <stdio.h>
#include <stdlib.h>

struct node{
    int val;
    struct node *left, *right;
};
typedef struct node *NODE;
NODE root=NULL;
int opcount = 0;

NODE insert(int ele, NODE node){
    NODE temp;
    if(node == NULL){
        temp= (NODE)malloc(sizeof(struct node));
        temp->val=ele;
        temp->left = temp->right=NULL;
        if(root == NULL)root=temp;
        return temp;
    }else if(ele < node->val){
        node->left = insert(ele, node->left);
        return node;
    }else if(ele > node->val){
        node->right = insert(ele,node->right);
        return node;
    }else {
        printf("Duplicate element found while inserting. Insertion failed\n");
        return NULL;
    }
}

int node_count(NODE node){
    opcount++;
    if(node==NULL)return 0;
    else return node_count(node->left) + node_count(node->right) + 1;
}

void main() {
    int choice,ele;
    printf("1. Insert an element\n");
    printf("2. Find Height of BST\n");
    printf("3. Exit\n");
    do{
        printf("Please enter your choice: ");
        scanf("%d",&choice);
        switch(choice) {
            case 1:
                printf("    Please enter an element: ");
                scanf("%d", &ele);
                insert(ele,root);
        break;
            case 2:
                opcount = 0;
                printf("    HNumber of Nodes : %d\n",node_count(root));
                printf("    Opcount=%d \n",opcount);
        break;
```

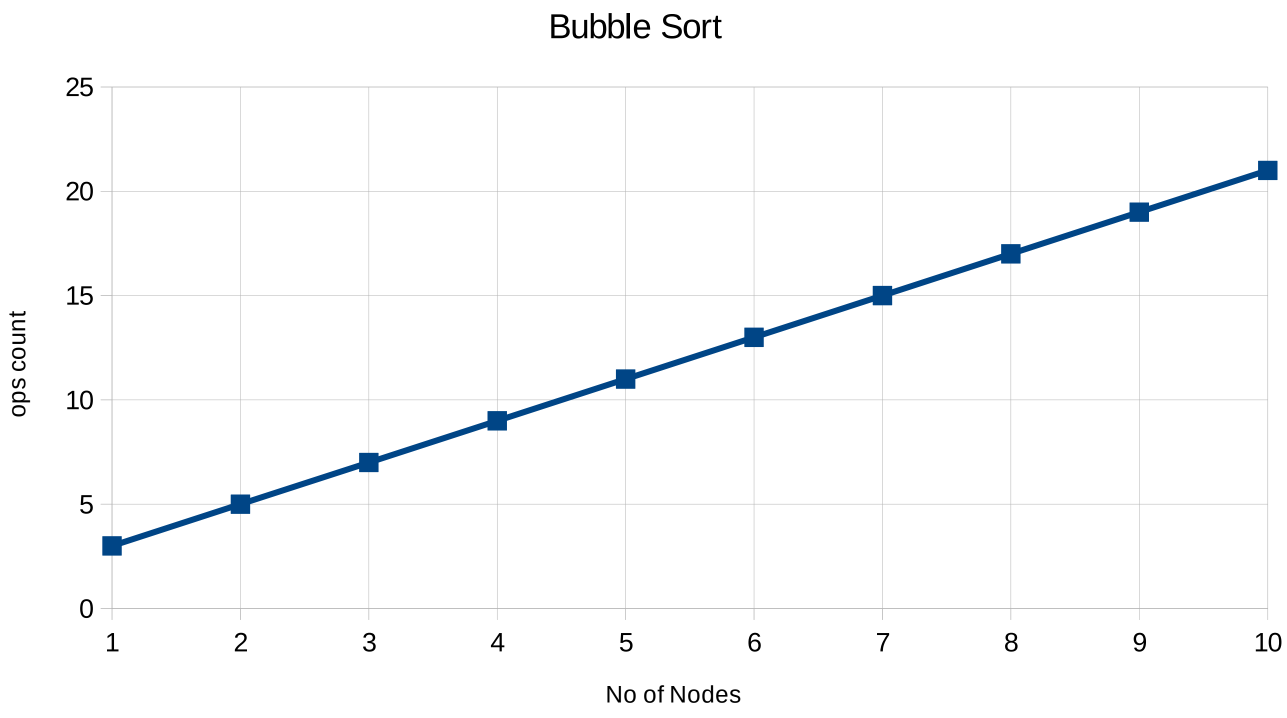
```
            case 3:
                break;
            default: printf("Invalid choice. Please enter valid choice\n");
        break;
        }
    }while(choice != 3);
}
```

Sample Input/Output:



```
student@lpcp-23:~/Desktop/230962326/Week6$ ./q1
1. Insert an element
2. Find No of Nodes of BST
3. Exit
Please enter your choice: 1
    Please enter an element: 3
Please enter your choice: 1
    Please enter an element: 5
Please enter your choice: 1
    Please enter an element: 80
Please enter your choice: 1
    Please enter an element: 55
Please enter your choice: 2
    HNumber of Nodes : 4
    Opcount=9
Please enter your choice: 3
student@lpcp-23:~/Desktop/230962326/Week6$
```

Graph:

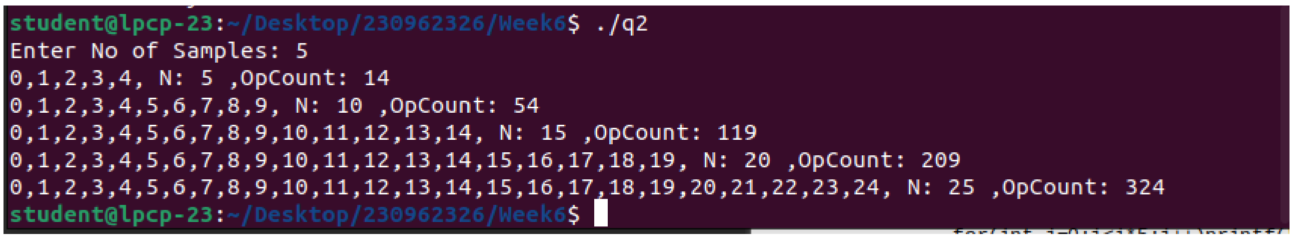


3) Sort given set of integers using Quick sort and analyze its efficiency. Obtain the experimental result of order of growth and plot the result.

Code:

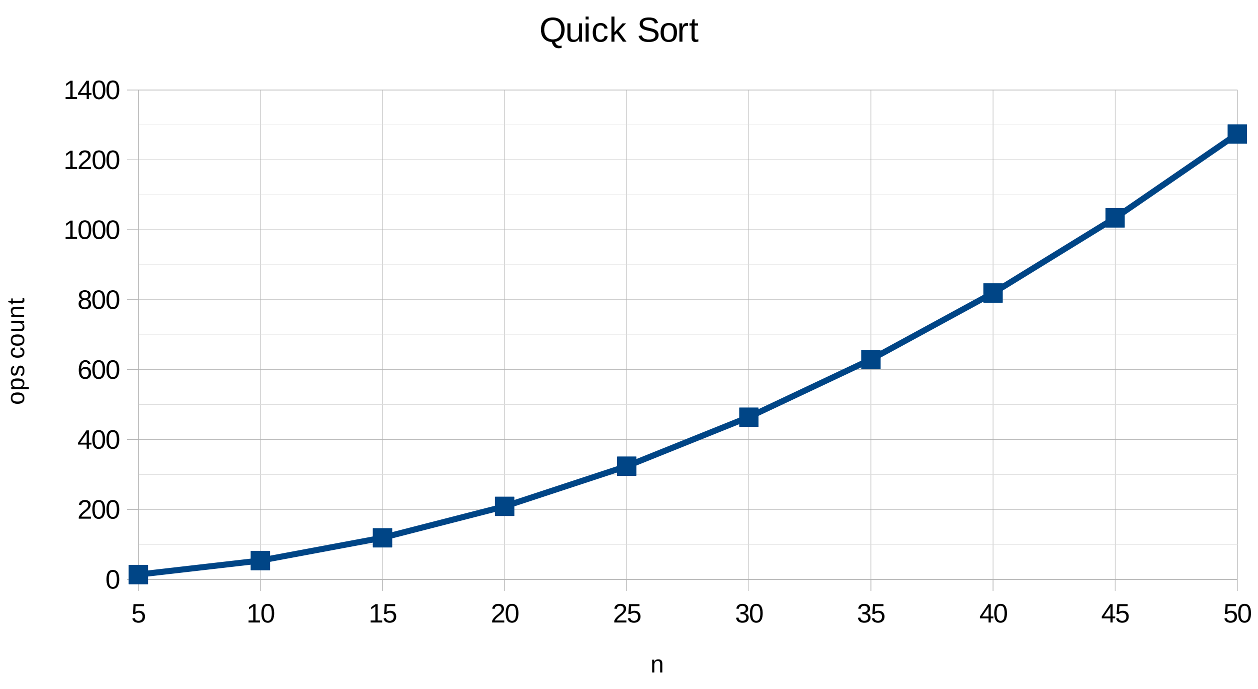
```c
#include <stdio.h>
#include <stdlib.h>

void swap(int* a, int* b) {
    int t = *a;
    *a = *b;
    *b = t;
}
int opcount = 0;
int partition(int arr[], int low, int high) {
    int pivot = arr[high];
    int i = low - 1;
    opcount++;
    for (int j = low; j <= high - 1; j++) {
        opcount++;
        if (arr[j] < pivot) {
            i++;
            swap(&arr[i], &arr[j]);
        }
    }
    swap(&arr[i + 1], &arr[high]);
    return i + 1;
}
void quickSort(int arr[], int low, int high) {
    if (low < high) {
        int pi = partition(arr, low, high);
        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}

int main() {
    int NoOfSamples;
    printf("Enter No of Samples: ");scanf("%d",&NoOfSamples);
    for(int i = 1;i <= NoOfSamples;i++){
        int arr[i*5];
        for(int j=0;j<i*5;j++)arr[j] = (i*5)-1-j;
        opcount = 0;
        quickSort(arr,0,(i*5)-1);
        for(int j=0;j<i*5;j++)printf("%d,",arr[j]);
        printf(" N: %d ,OpCount: %d\n",i*5,opcount);
    }
    return 0;
}
```

Sample Input/Output:



```
student@lpcp-23:~/Desktop/230962326/Week6$ ./q2
Enter No of Samples: 5
0,1,2,3,4, N: 5 ,OpCount: 14
0,1,2,3,4,5,6,7,8,9, N: 10 ,OpCount: 54
0,1,2,3,4,5,6,7,8,9,10,11,12,13,14, N: 15 ,OpCount: 119
0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19, N: 20 ,OpCount: 209
0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24, N: 25 ,OpCount: 324
student@lpcp-23:~/Desktop/230962326/Week6$
```

Graph:

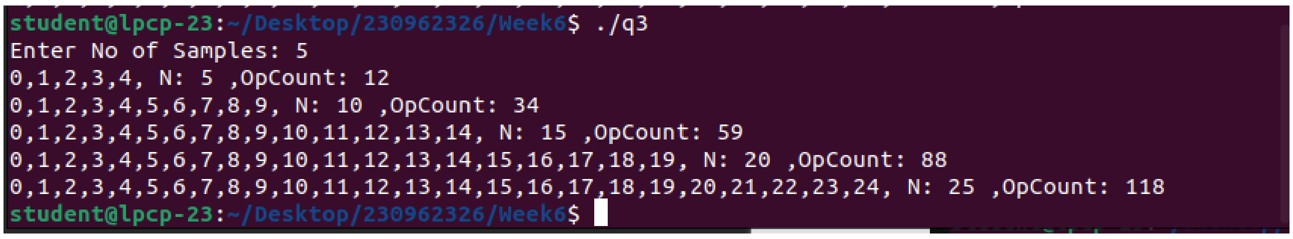


4) Sort the given set of integers using Merge sort and display the number of inversions performed during the merging step. Obtain the experimental result of order of growth by analysing its efficiency and plot the result.

Code:

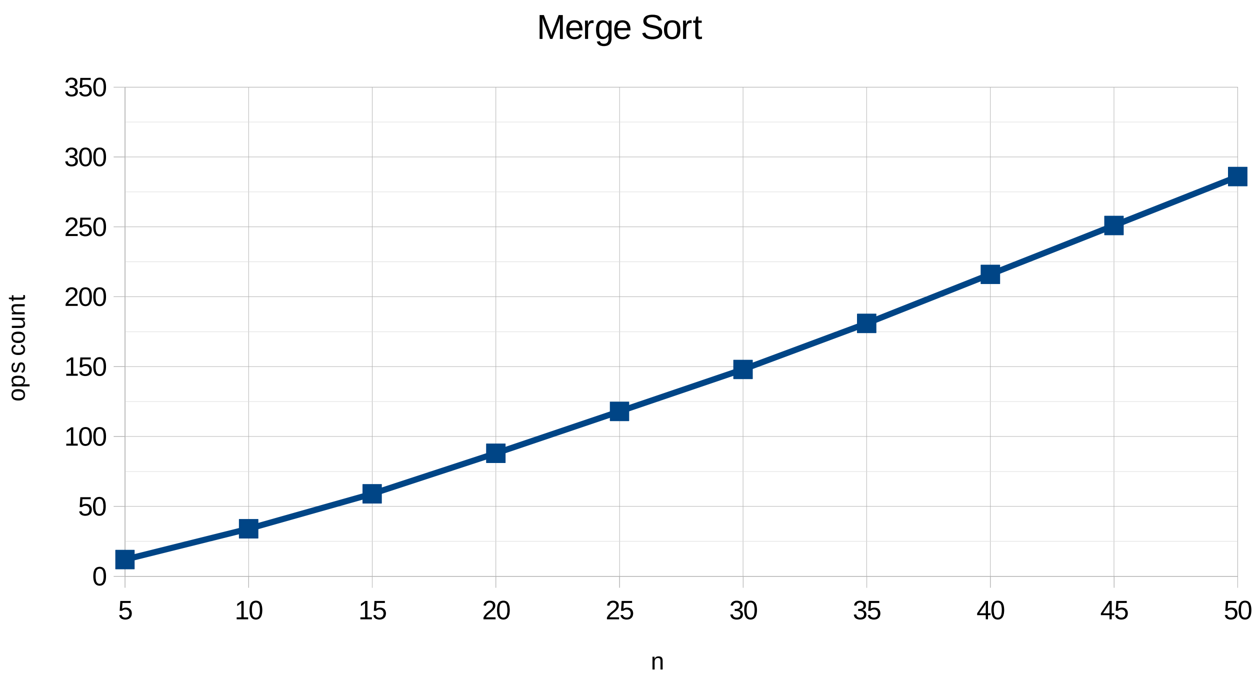
```
#include <stdio.h>
#include <stdlib.h>

int opcount = 0;

void merge(int arr[], int l, int m, int r){
    int n1 = m -l + 1;
    int n2 = r - m;
    int R[n1],S[n2];
    for(int i = 0;i<n1;i++)R[i]=arr[l+i];
    for(int j = 0;j<n2;j++)S[j]=arr[m+j+1];

    int i=0,j=0,k =0;
    while(i<n1 && j<n2){
        if(R[i] <= S[j]){
            arr[l+k] = R[i];
            i++;opcount++;
        }
        else{
            arr[l+k] = S[j];
            j++;opcount++;
        }
        k++;
    }
    while(i < n1){
        arr[l+k] = R[i];
        i++;k++;opcount++;
    }
    while(j < n2){
        arr[l+k] = S[i];
        j++;k++;opcount++;
    }
}

void mergesort(int arr[],int l,int r){
    if(l < r){
        int m = l + (r-l)/2;
        mergesort(arr,l,m);
        mergesort(arr,m+1,r);

        merge(arr,l,m,r);
    }
}

int main() {
    int NoOfSamples;
    printf("Enter No of Samples: ");scanf("%d",&NoOfSamples);
    for(int i = 1;i <= NoOfSamples;i++){
        int arr[i*5];
        for(int j=0;j<i*5;j++)arr[j] = (i*5)-1-j;
        opcount = 0;
        mergesort(arr,0,(i*5)-1);
        for(int j=0;j<i*5;j++)printf("%d,",arr[j]);
        printf(" N: %d ,OpCount: %d\n",i*5,opcount);
    }
    return 0;
}
```

Sample Input/Output:



```
student@lpcp-23:~/Desktop/230962326/Week6$ ./q3
Enter No of Samples: 5
0,1,2,3,4, N: 5 ,OpCount: 12
0,1,2,3,4,5,6,7,8,9, N: 10 ,OpCount: 34
0,1,2,3,4,5,6,7,8,9,10,11,12,13,14, N: 15 ,OpCount: 59
0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19, N: 20 ,OpCount: 88
0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24, N: 25 ,OpCount: 118
student@lpcp-23:~/Desktop/230962326/Week6$
```

Graph:


Merge Sort
ops count
n