MAULANA AZAD NATIONAL INSTITUTE OF TECHNOLOGY

Name: Vivek Kumar Ahirwar **Scholar No:** 191112419

Department: CSE **Section:**3

Semester: 4th **Subject:** ADA Lab

Date: 20/01/2020

Subject	**Analysis and Design of Algorithms LAB**	**Session: Jan 2021**
Sub. Code:	**CSE-228**	**Semester: IV (CSE)**
Name of Teacher	**Prof. Manish Pandey**	

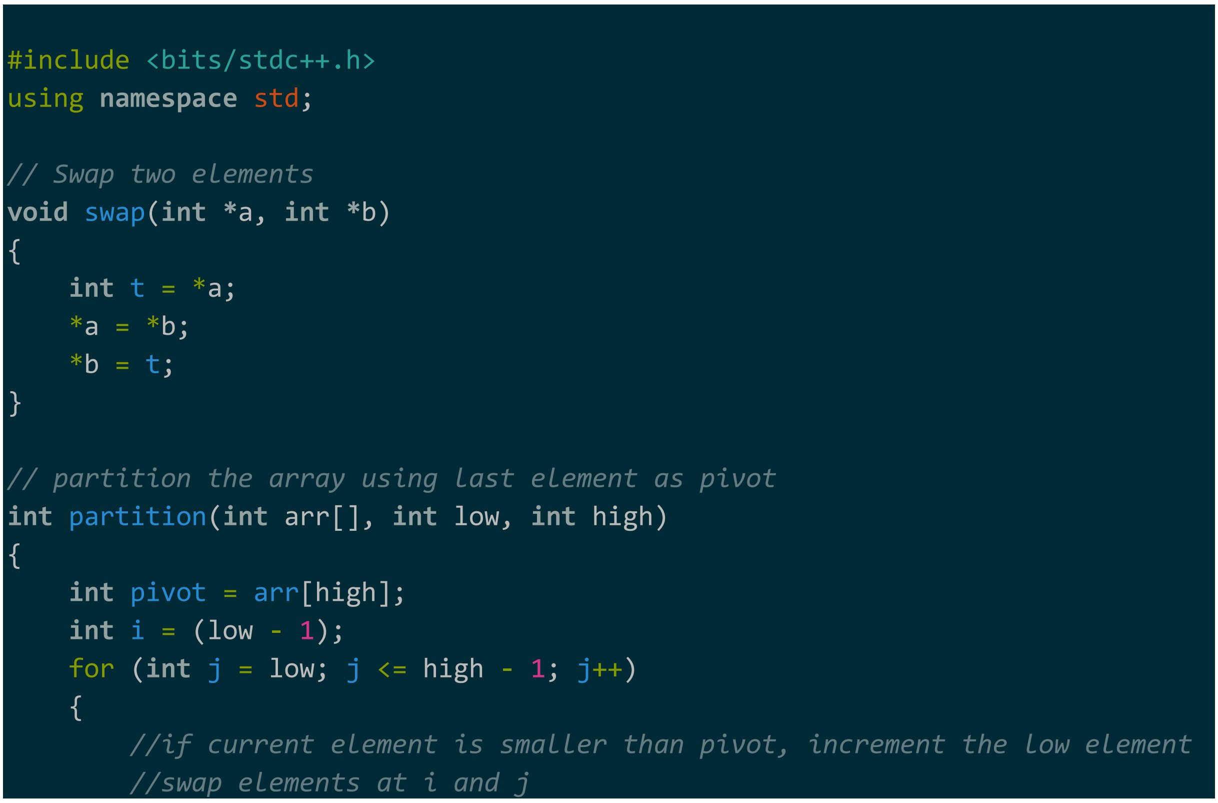
ADA: LAB-ASSIGNMENT 2

QUE: Write programs for implementing the following sorting techniques to arrange a list of integers in ascending order. Also, analyze the time complexity:

a) Quick Sort

b) Merge Sort

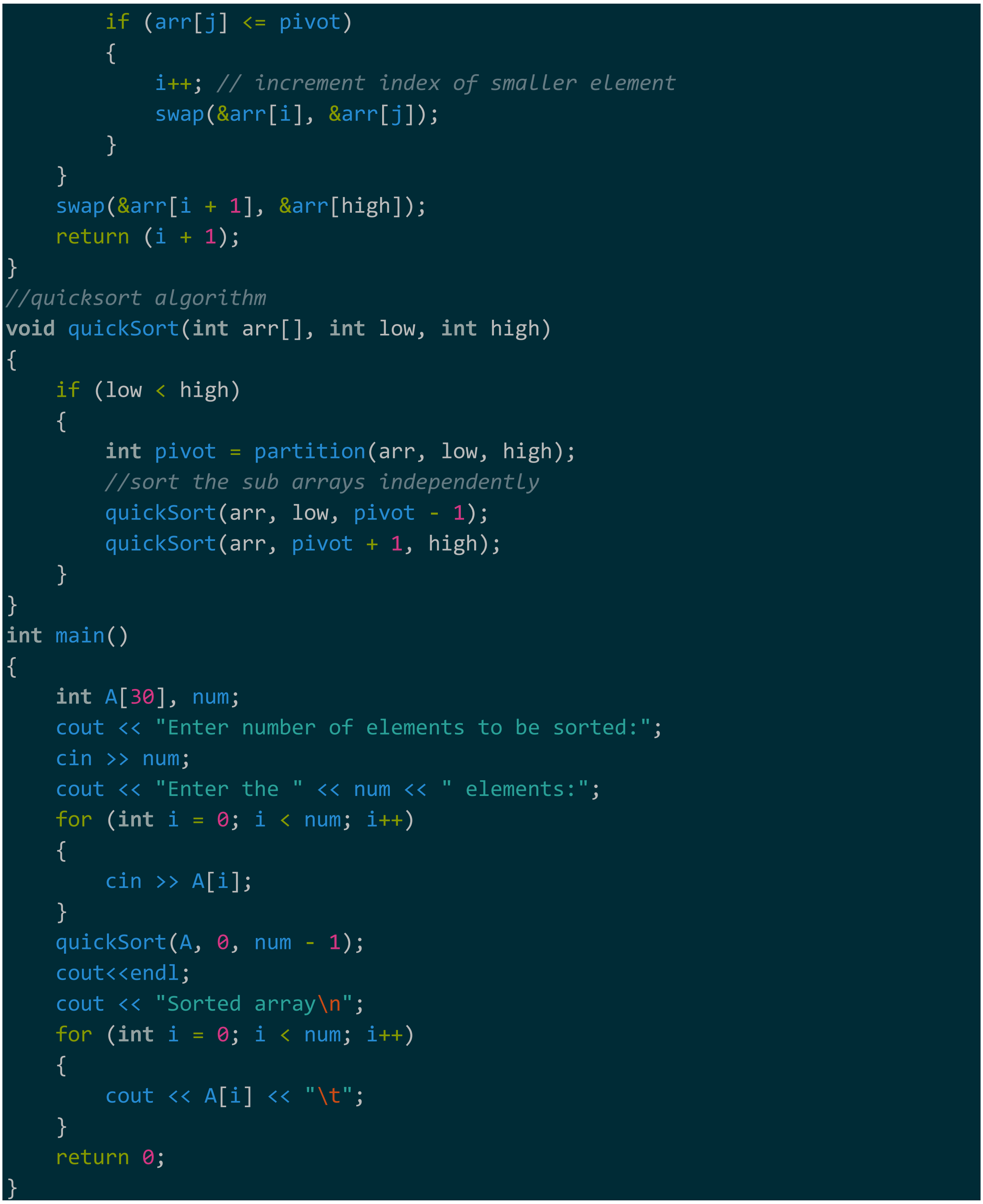
Quick Sort



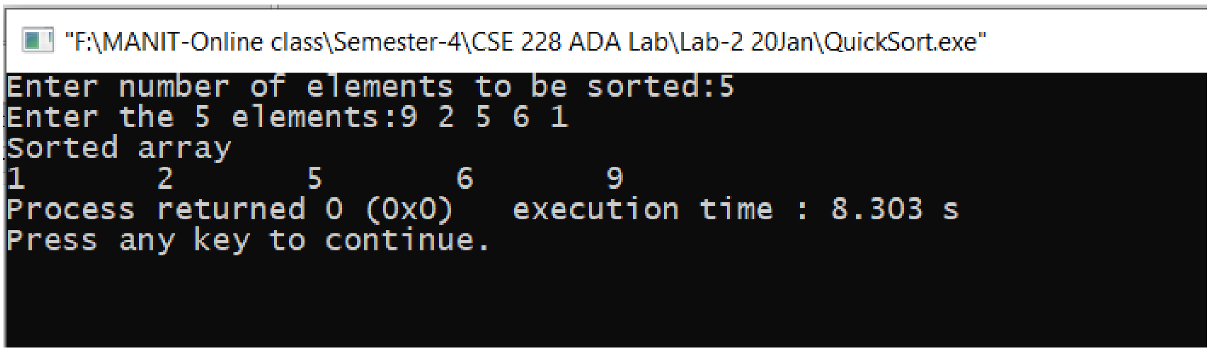
```
#include <bits/stdc++.h>
using namespace std;

// Swap two elements
void swap(int *a, int *b)
{
    int t = *a;
    *a = *b;
    *b = t;
}

// partition the array using last element as pivot
int partition(int arr[], int low, int high)
{
    int pivot = arr[high];
    int i = (low - 1);
    for (int j = low; j <= high - 1; j++)
    {
        //if current element is smaller than pivot, increment the low element
        //swap elements at i and j
```



```
        if (arr[j] <= pivot)
        {
            i++; // increment index of smaller element
            swap(&arr[i], &arr[j]);
        }
    }
    swap(&arr[i + 1], &arr[high]);
    return (i + 1);
}
//quicksort algorithm
void quickSort(int arr[], int low, int high)
{
    if (low < high)
    {
        int pivot = partition(arr, low, high);
        //sort the sub arrays independently
        quickSort(arr, low, pivot - 1);
        quickSort(arr, pivot + 1, high);
    }
}
int main()
{
    int A[30], num;
    cout << "Enter number of elements to be sorted:";
    cin >> num;
    cout << "Enter the " << num << " elements:";
    for (int i = 0; i < num; i++)
    {
        cin >> A[i];
    }
    quickSort(A, 0, num - 1);
    cout<<endl;
    cout << "Sorted array\n";
    for (int i = 0; i < num; i++)
    {
        cout << A[i] << "\t";
    }
    return 0;
}
```



"F:\MANIT-Online class\Semester-4\CSE 228 ADA Lab\Lab-2 20Jan\QuickSort.exe"

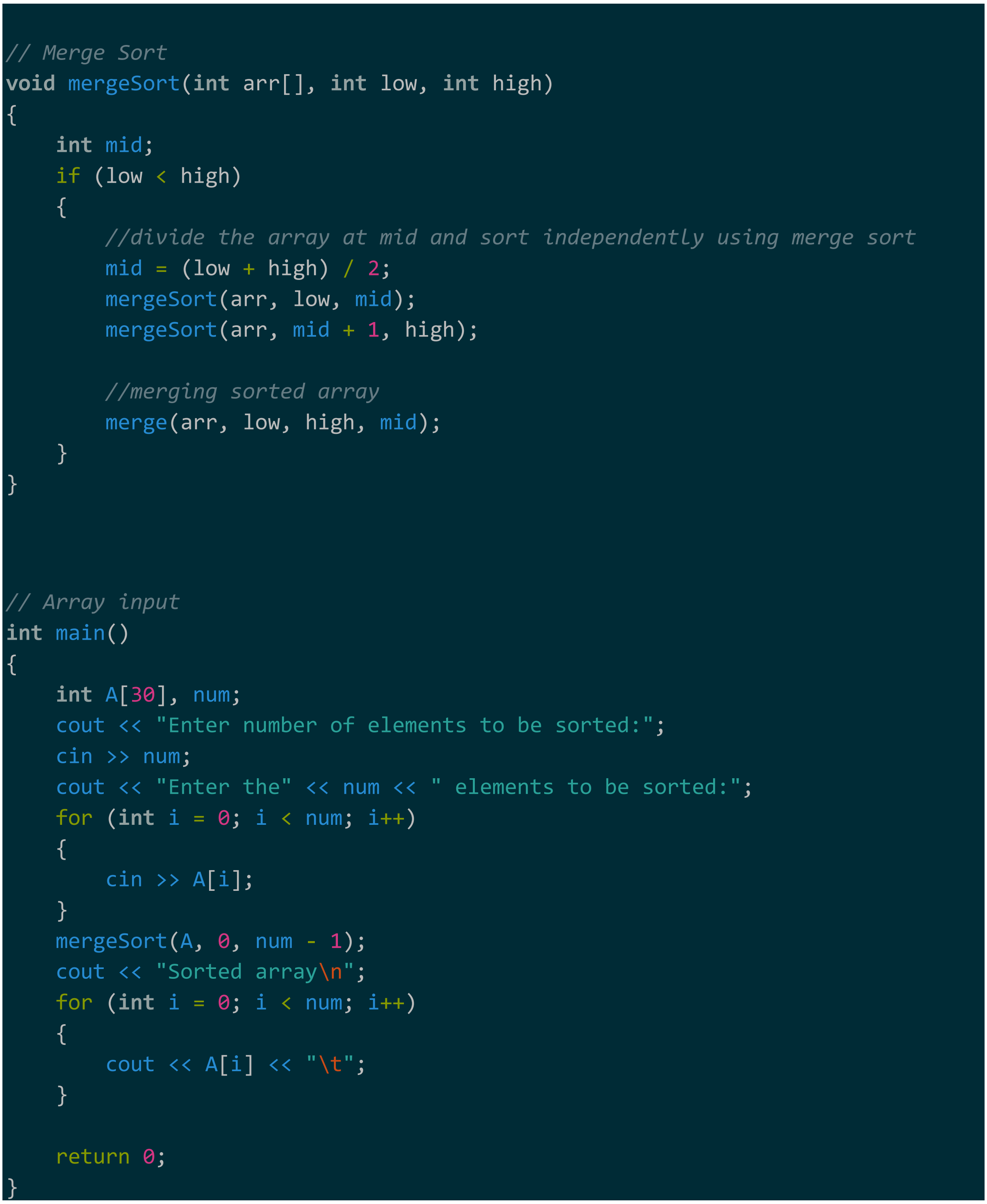
```
Enter number of elements to be sorted:5
Enter the 5 elements:9 2 5 6 1
Sorted array
1       2       5       6       9
Process returned 0 (0x0)   execution time : 8.303 s
Press any key to continue.
```

Merge Sort

```
//Merge Sort

#include <bits/stdc++.h>
using namespace std;


// Function for Merging
void merge(int arr[], int low, int high, int mid)
{
    int i, j, k, c[50];
    i = low;
    k = low;
    j = mid + 1;
    while (i <= mid && j <= high)
    {
        if (arr[i] < arr[j])
        {
            c[k] = arr[i];
            k++;
            i++;
        }
        else
        {
            c[k] = arr[j];
            k++;
            j++;
        }
    }
    while (i <= mid)
    {
        c[k] = arr[i];
        k++;
        i++;
    }
    while (j <= high)
    {
        c[k] = arr[j];
        k++;
        j++;
    }
    for (i = low; i < k; i++)
    {
        arr[i] = c[i];
    }
}
```



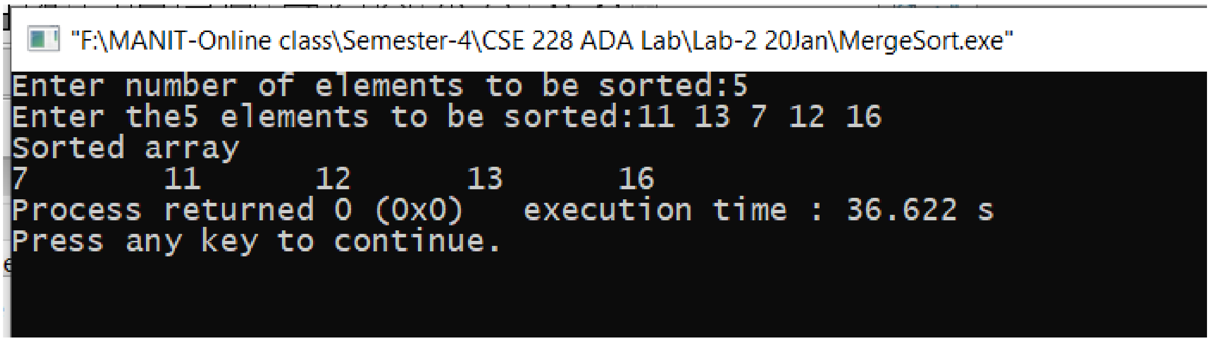
```cpp
// Merge Sort
void mergeSort(int arr[], int low, int high)
{
    int mid;
    if (low < high)
    {
        //divide the array at mid and sort independently using merge sort
        mid = (low + high) / 2;
        mergeSort(arr, low, mid);
        mergeSort(arr, mid + 1, high);

        //merging sorted array
        merge(arr, low, high, mid);
    }
}



// Array input
int main()
{
    int A[30], num;
    cout << "Enter number of elements to be sorted:";
    cin >> num;
    cout << "Enter the" << num << " elements to be sorted:";
    for (int i = 0; i < num; i++)
    {
        cin >> A[i];
    }
    mergeSort(A, 0, num - 1);
    cout << "Sorted array\n";
    for (int i = 0; i < num; i++)
    {
        cout << A[i] << "\t";
    }

    return 0;
}
```



"F:\MANIT-Online class\Semester-4\CSE 228 ADA Lab\Lab-2 20Jan\MergeSort.exe"

```
Enter number of elements to be sorted:5
Enter the5 elements to be sorted:11 13 7 12 16
Sorted array
7       11      12      13      16
Process returned 0 (0x0)   execution time : 36.622 s
Press any key to continue.
```