**EXPERIMENT-5**

**AIM:** Quick Sort with its time complexity.

**CODE:**

#include <iostream>

using namespace std;

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

    int\* temp = new int(1);

    temp = a;

    a = b;

    b = temp;

    free(temp);

}

int partition(int A[], int low, int high){

    int pivot = A[low];

    int i = low, j = high;

    do{

        do{i++;}while(A[i]<=pivot);

        do{j--;}while (A[j]>pivot);

        if (i<j) swap(A[i], A[j]);

    }while (i<j);

    swap(A[low], A[j]);

    return j;

}

void QuickSort(int A[], int low, int high){

    int j;

    if(low<high){

        j = partition(A, low, high);

        QuickSort(A, low, j);

        QuickSort(A, j+1, high);

    }

}

int main(){

*// int A[] = {11,13,7,12,16,9,24,5,10,3,677},n=11,i;*

    int A[] = {11,13,7,12,16,9,24,5,10,3,INT32\_MAX},n=11,i;

    int \*B = new int(11);

    B = A;

    QuickSort(B, 0, 10);

    for(i = 0; i<10; i++){

        cout<<B[i]<<" ";

    }

    cout<<endl;

    return 0;

}

Time complexity:

Best case: O(n log n)

Worst case: O(n)2

General case: O(n log n)