

DESIGN AND ANALYSIS OF ALGORITHMS LAB (PCS-409)

WEEK 3

1 . Given an unsorted array of integers, design an algorithm and a program to sort the array using insertion sort. Your program should be able to find number of comparisons and shifts (shifts - total number of times the array elements are shifted from their place) required for sorting the array.

```
#include <iostream>
#include <vector>
#include <fstream>
using namespace std;

                                                                    //function
void insertionSort(vector<int>& arr,ofstream &fout) {
    int comparisons = 0, shifts = 0;
    for(int i = 1; i < arr.size(); i++) {
        int j = i - 1;
        int temp = arr[i];
        while(j >= 0 && temp < arr[j]) {
            comparisons++;
            shifts++;
            arr[j + 1] = arr[j];
            j--;
        }
        shifts++;
        arr[j + 1] = temp;
    }
    for(int i = 0; i < arr.size(); i++) {
        fout<<arr[i]<<" ";
    }
    fout<<endl;
    fout<<"comparisons = "<<comparisons<<endl;
    fout<<"shifts = "<<shifts<<endl;
}
```

DESIGN AND ANALYSIS OF ALGORITHMS LAB (PCS-409)

```
int main() {  
  
                                                                    // Open input and output files  
  
    ifstream fin("input.txt");  
    ofstream fout("output.txt");  
    // Check if files are opened successfully  
    if (!fin.is_open() || !fout.is_open()) {  
        cout << "Error occurred while opening files.\n";  
        return 0;  
    }  
    int t;  
  
                                                                    // Number of test cases  
  
    fin >> t;  
    while (t--> 0) {  
  
        int n;  
        fin >> n;  
        vector<int> arr(n);  
  
                                                                    // Input array elements  
  
        for (int i = 0; i < n; i++) {  
            fin >> arr[i];  
        }  
  
                                                                    // Perform insertion sort  
  
        insertionSort(arr,fout);  
    }  
  
                                                                    // Close input and output files  
  
    fin.close();  
    fout.close();  
}
```

DESIGN AND ANALYSIS OF ALGORITHMS LAB (PCS-409)

main.cpp	input.txt	:	output.txt	:
1	3			
2	8			
3	-23 65 -31 76 46 89 45 32			
4	10			
5	54 65 34 76 78 97 46 32 51 21			
6	15			
7	63 42 223 645 652 31 324 22 553 -12 54 65 86 46 325			

DESIGN AND ANALYSIS OF ALGORITHMS LAB (PCS-409)

main.cpp	input.txt	:	output.txt	:
1	-31 -23 32 45 46 65 76 89			
2	comparisons = 13			
3	shifts = 20			
4	21 32 34 46 51 54 65 76 78 97			
5	comparisons = 28			
6	shifts = 37			
7	-12 22 31 42 46 54 63 65 86 223 324 325 553 645 652			
8	comparisons = 54			
9	shifts = 68			
10				

DESIGN AND ANALYSIS OF ALGORITHMS LAB (PCS-409)

2 . Given an unsorted array of integers, design an algorithm and implement a program to sort this array using selection sort. Your program should also find number of comparisons and number of swaps required.

```
#include <iostream>
#include <vector>
#include <fstream>
using namespace std;

                                                                    //function
void selectionSort(vector<int>& arr,ofstream &fout) {
    int smallest, swaps = 0, comparisons = 0;
    for(int i = 0; i < arr.size() - 1; i++) {
        smallest = i;
        for(int j = i + 1; j < arr.size(); j++) {
            comparisons++;
            if(arr[smallest] > arr[j]) {
                smallest = j;
            }
        }
        swaps++;
        int temp = arr[smallest];
        arr[smallest] = arr[i];
        arr[i] = temp;
    }
    for(int i = 0; i < arr.size(); i++)
        fout<<arr[i]<<" ";
    fout<<endl;
    fout<<"comparisons = "<<comparisons<<endl;
    fout<<"swaps = "<<swaps<<endl;
}
```

DESIGN AND ANALYSIS OF ALGORITHMS LAB (PCS-409)

```
int main() {  
  
                                                                    // Open input and output files  
    ifstream fin("input.txt");  
    ofstream fout("output.txt");  
    // Check if files are opened successfully  
    if (!fin.is_open() || !fout.is_open()) {  
        cout << "Error occurred while opening files.\n";  
        return 0;  
    }  
    int t;  
  
                                                                    // Number of test cases  
    fin >> t;  
    while (t--> 0) {  
  
        int n;  
        fin >> n;  
        vector<int> arr(n);  
  
                                                                    // Input array elements  
        for (int i = 0; i < n; i++) {  
            fin >> arr[i];  
        }  
  
                                                                    // Perform selection sort  
        selectionSort(arr,fout);  
    }  
    // Close input and output files  
    fin.close();  
    fout.close();  
}
```

DESIGN AND ANALYSIS OF ALGORITHMS LAB (PCS-409)

main.cpp	input.txt	output.txt
1	3	
2	8	
3	-23 65 -31 76 46 89 45 32	
4	10	
5	54 65 34 76 78 97 46 32 51 21	
6	15	
7	63 42 223 645 652 31 324 22 553 -12 54 65 86 46 325	

DESIGN AND ANALYSIS OF ALGORITHMS LAB (PCS-409)

main.cpp	input.txt	:	output.txt	:
1	-31 -23 32 45 46 65 76 89			
2	comparisons = 28			
3	swaps = 7			
4	21 32 34 46 51 54 65 76 78 97			
5	comparisons = 45			
6	swaps = 9			
7	-12 22 31 42 46 54 63 65 86 223 324 325 553 645 652			
8	comparisons = 105			
9	swaps = 14			
10				

DESIGN AND ANALYSIS OF ALGORITHMS LAB (PCS-409)

3 . . Given an unsorted array of positive integers, design an algorithm and implement it using a program to find whether there are any duplicate elements in the array or not. (use sorting) (Time Complexity = $O(n \log n)$)

```
#include <iostream>
#include <vector>
#include <fstream>
using namespace std;

//merge function

void merge(vector<int>&nums,int l,int mid,int r){
    int n1 = mid-l+1 , n2 = r-mid;
    int a1[n1],a2[n2];
    for(int i=0;i<n1;i++){
        a1[i]=nums[i+l];
    }
    for(int i=0;i<n2;i++){
        a2[i] = nums[i+mid+1];
    }
    int j=0,k=0;
    for(int i=l;i<=r;i++){
        if( j <n1 && k<n2){
            if(a1[j]<a2[k]){
                nums[i] = a1[j];
                j++;
            }else{
                nums[i]=a2[k];
                k++;
            }
        }
        else if(j<n1){
            nums[i] = a1[j];
            j++;
        }
    }
}
```

DESIGN AND ANALYSIS OF ALGORITHMS LAB (PCS-409)

```
    }
    else{
        nums[i]=a2[k];
        k++;
    }
}

}

}

//mergeSort (nlogn)

void mergeSort(vector<int>&nums,int l,int r){
    if(l>=r) return;
    int mid = (l+r)/2;
    mergeSort(nums,l,mid);
    mergeSort(nums,mid+1,r);
    merge(nums,l,mid,r);
}

//function to find duplicates

void duplicates(vector<int>& arr,ofstream &fout) {
    mergeSort(arr,0,arr.size()-1);
    int i = 0;
    while(i < arr.size() - 1) {
        if(arr[i] == arr[i + 1]) {
            fout<<"YES"<<endl;
            return;
        }
        i++;
    }
    fout<<"NO"<<endl;
}
```

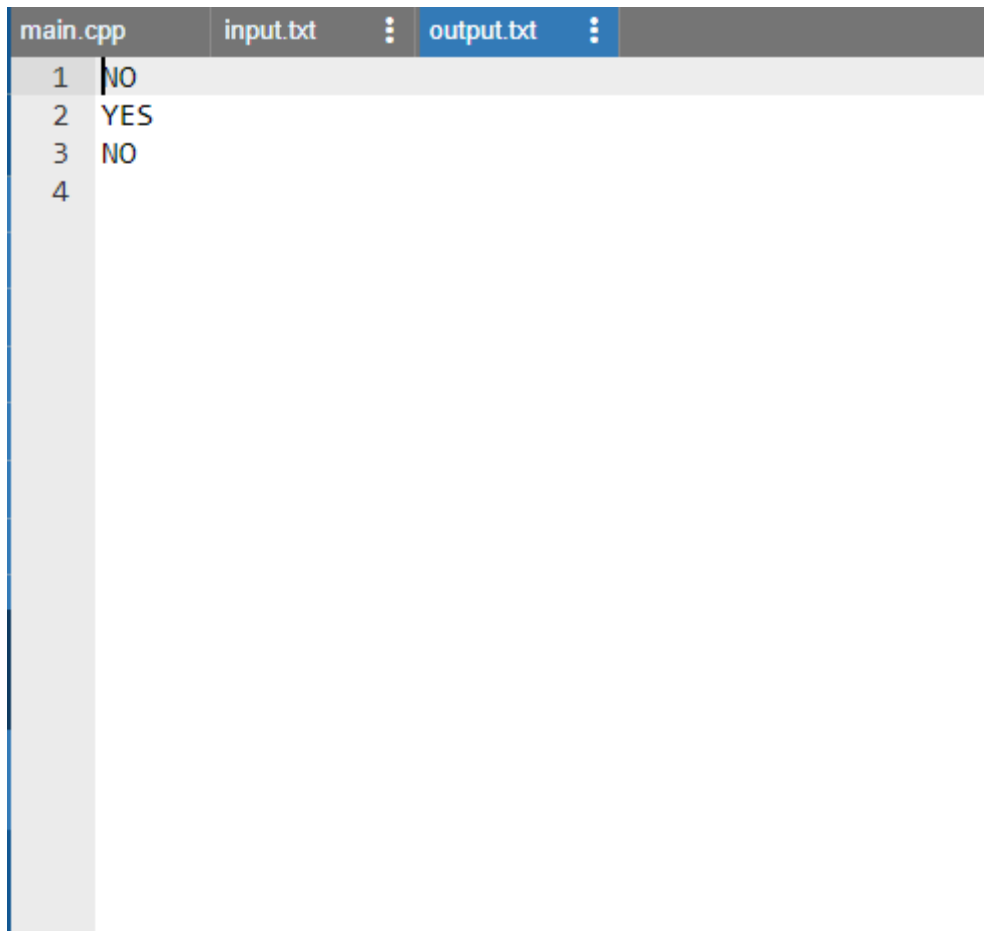
DESIGN AND ANALYSIS OF ALGORITHMS LAB (PCS-409)

```
int main() {  
  
                                                                    // Open input and output files  
    ifstream fin("input.txt");  
    ofstream fout("output.txt");  
  
                                                                    // Check if files are opened successfully  
    if (!fin.is_open() || !fout.is_open()) {  
        cout << "Error occurred while opening files.\n";  
        return 0;  
    }  
    int t;  
  
                                                                    // Number of test cases  
    fin >> t;  
    while (t--> 0) {  
  
        int n;  
        fin >> n;  
        vector<int> arr(n);  
  
                                                                    // Input array elements  
        for (int i = 0; i < n; i++) {  
            fin >> arr[i];  
        }  
  
                                                                    // finding suplicates  
        duplicates(arr,fout);  
    }  
  
                                                                    // Close input and output files  
    fin.close();  
    fout.close();  
}
```

DESIGN AND ANALYSIS OF ALGORITHMS LAB (PCS-409)

main.cpp	input.txt	:	output.txt	:
1	3			
2	5			
3	28 52 83 14 75			
4	10			
5	75 65 1 65 2 6 86 2 75 8			
6	15			
7	75 35 86 57 98 23 73 1 64 8 11 90 61 19 20			

DESIGN AND ANALYSIS OF ALGORITHMS LAB (PCS-409)



```
main.cpp  input.txt  output.txt
1 NO
2 YES
3 NO
4
```