

***RAJSHAHI UNIVERSITY OF ENGINEERING & TECHNOLOGY***



***DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING***

**Lab Report 7**

**Course Code:** CSE 2202

**Course Title:** Sessional Based on CSE 2201.

**Submitted By:**

Name : Shanjid Hasan Nishat

Roll No : 1803172

Section : 'C'

Date of Submission: 03/06/ 2021

**Submitted To:**

Biprodip Pal

Assistant Professor,

Dept. of Computer Science and

Engineering.

Rajshahi University of

Engineering & Technology.

**Problem Statement:** Sorting in linear time complexity with **Counting Sort**.

**Code:**

### Random Input Generation:

```
#include <bits/stdc++.h>
using std::cerr;
using std::endl;
#include <fstream>
using std::ofstream;
#include <cstdlib>
int main()
{
    ofstream outdata;
    int i;

    outdata.open("input.txt");
    if( !outdata )
    {
        cerr << "Error: file could not be opened" << endl;
        exit(1);
    }
    for (i=0; i<50; ++i)
    {
        int x = rand();
        outdata << x%10001 << endl;
    }
    outdata.close();

    outdata.open("input_100.txt");
    if( !outdata )
    {
        cerr << "Error: file could not be opened" << endl;
        exit(1);
    }
    for (i=0; i<100; ++i)
    {
        int x = rand();
        outdata << x%10001 << endl;
    }
    outdata.close();
    outdata.open("input_500.txt");
    if( !outdata )
    {
        cerr << "Error: file could not be opened" << endl;
        exit(1);
    }
    for (i=0; i<500; ++i)
    {
        int x = rand();
        outdata << x%10001 << endl;
    }
    outdata.close();
    outdata.open("input_1000.txt");
```

```

if( !outdata )
{
    cerr << "Error: file could not be opened" << endl;
    exit(1);
}
for (i=0; i<1000; ++i)
{
    int x = rand();
    outdata << x%10001 << endl;
}
outdata.close();
outdata.open("input_1500.txt");
if( !outdata )
{
    cerr << "Error: file could not be opened" << endl;
    exit(1);
}
for (i=0; i<1500; ++i)
{
    int x = rand();
    outdata << x%10001 << endl;
}
outdata.close();

outdata.open("input_2000.txt");
if( !outdata )
{
    cerr << "Error: file could not be opened" << endl;
    exit(1);
}
for (i=0; i<2000; ++i)
{
    int x = rand();
    outdata << x << endl;
}
outdata.close();
return 0;
}

```

## Sorting numbers using Bubble Sort and Counting Sort:

```
#include <bits/stdc++.h>
using namespace std;
long long arr[100010], len = 0;
long long cnt1=0, cnt2=0;
void readFile(string fname)
{
    long long x, i=0;
    ifstream inFile;
    inFile.open(fname);
    if (!inFile)
    {
        cout << "Cannot open file.\n";
        exit(1);
    }
    while (inFile >> x)
    {
        arr[i++] = x;
    }
    inFile.close();
    len = i;
}
void find_max(long long i, long long n, long long &maxi, long long &cnt2)
{
    cnt2++;
    if (n-i == 0)
    {
        cnt2+=1;
        maxi = max(arr[i], maxi);
        return;
    }
    else if (n-i == 1)
    {
        cnt2+=1;
        maxi = max(maxi, max(arr[i], arr[n]));
        return;
    }
    cnt2+=2;
    find_max(i, (i+n)/2, maxi, cnt2);
    find_max(((i+n)/2+1), n, maxi, cnt2);
}
void bubble_sort(long long a[], long long n)
{
    long long temp, i;
    for (i = 0; i < n-1; i++)
    {
        cnt1+=2;
        for (int j = 0; j < n-i-1; j++)
        {
            cnt1+=2;
            cnt1++;
            if (a[j] > a[j+1] )
            {
                cnt1+=3;
                temp = a[j];
```

```

        a[j] = a[j+1];
        a[j+1] = temp;
    }
}

cout<<"Bubble sort : \n";
for(i = 0 ; i < n ; i++)
{
    cout<<a[i]<<"\t";
}

}

void countSort(long long a[], long long n)
{
    long long output[n+1];
    long long maxi = LLONG_MIN, i;
    find_max(0, n-1, maxi, cnt2);
    long long count_arr[maxi+1];
    for(i = 0; i<=maxi; i++)
    {cnt2+=2;
        count_arr[i] = 0;cnt2++;
    }

    for(i = 0; i<n; i++)
    {cnt2+=2;
        count_arr[a[i]]++;cnt2++;
    }
    for(i = 1; i<=maxi; i++)
    {cnt2+=2;
        count_arr[i] += count_arr[i-1];cnt2++;
    }
    for(i = n - 1; i>=0 ; i--)
    {cnt2+=2;
        output[--count_arr[a[i]]] = a[i];cnt2++;
    }
    cout<<"\n\nCounting sort : \n";
    for(i = 0 ; i < n ; i++)
    {
        cout<<output[i]<<"\t";
    }
}

int main()
{
    cnt1 = 0;
    cnt2 = 0;
    readFile("input.txt");
    bubble_sort(arr, len);
    readFile("input.txt");
    countSort(arr, len);
    cout<<"\n\nFor "<<len<<" data : \nStep count in Bubble Sort :
"<<cnt1<<"\nStep count in Counting Sort : "<<cnt2<<endl;

    cnt1 = 0;
    cnt2 = 0;
    readFile("input_100.txt");
    bubble_sort(arr, len);

```

```

        readFile("input_100.txt");
        countSort(arr, len);
        cout<<"\n\nFor "<<len<<" data : \nStep count in Bubble Sort :
"<<cnt1<<"\nStep count in Counting Sort : "<<cnt2<<endl;

        cnt1 = 0;
        cnt2 = 0;
        readFile("input_500.txt");
        bubble_sort(arr, len);
        readFile("input_500.txt");
        countSort(arr, len);
        cout<<"\n\nFor "<<len<<" data : \nStep count in Bubble Sort :
"<<cnt1<<"\nStep count in Counting Sort : "<<cnt2<<endl;

        cnt1 = 0;
        cnt2 = 0;
        readFile("input_1000.txt");
        bubble_sort(arr, len);
        readFile("input_1000.txt");
        countSort(arr, len);
        cout<<"\n\nFor "<<len<<" data : \nStep count in Bubble Sort :
"<<cnt1<<"\nStep count in Counting Sort : "<<cnt2<<endl;

        cnt1 = 0;
        cnt2 = 0;
        readFile("input_1500.txt");
        bubble_sort(arr, len);
        readFile("input_1500.txt");
        countSort(arr, len);
        cout<<"\n\nFor "<<len<<" data : \nStep count in Bubble Sort :
"<<cnt1<<"\nStep count in Counting Sort : "<<cnt2<<endl;

        cnt1 = 0;
        cnt2 = 0;
        readFile("input_2000.txt");
        bubble_sort(arr, len);
        readFile("input_2000.txt");
        countSort(arr, len);
        cout<<"\n\nFor "<<len<<" data : \nStep count in Bubble Sort :
"<<cnt1<<"\nStep count in Counting Sort : "<<cnt2<<endl;
        return 0;
}

```

## Output:

```
"F:\2-2\Study Materials\Sessional Based on CSE 2201\L6 Re...
Bubble sort :
41 153 292 330 491 1319 1477 1537 172
4 1869 1941 2381 2388 2995 3279 3809 390
2 4462 4603 4664 4770 4827 5140 5436 544
7 5665 5705 5723 6297 6334 6498 6826 686
8 6960 7034 7420 7672 7711 8143 8251 846
6 8701 8715 9168 9356 9717 9894 9894 991
1 9961

Counting sort :
41 153 292 330 491 1319 1477 1537 172
4 1869 1941 2381 2388 2995 3279 3809 390
2 4462 4603 4664 4770 4827 5140 5436 544
7 5665 5705 5723 6297 6334 6498 6826 686
8 6960 7034 7420 7672 7711 8143 8251 846
6 8701 8715 9168 9356 9717 9894 9894 991
1 9961

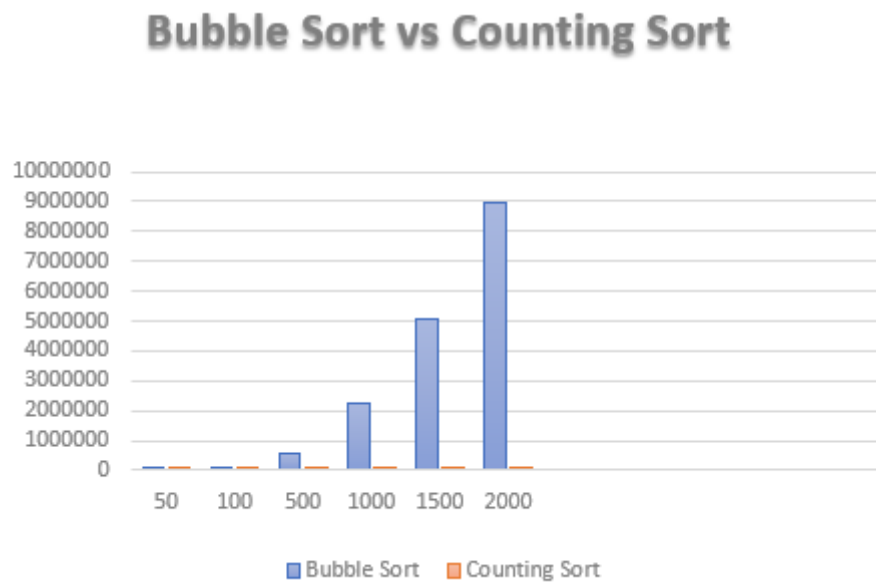
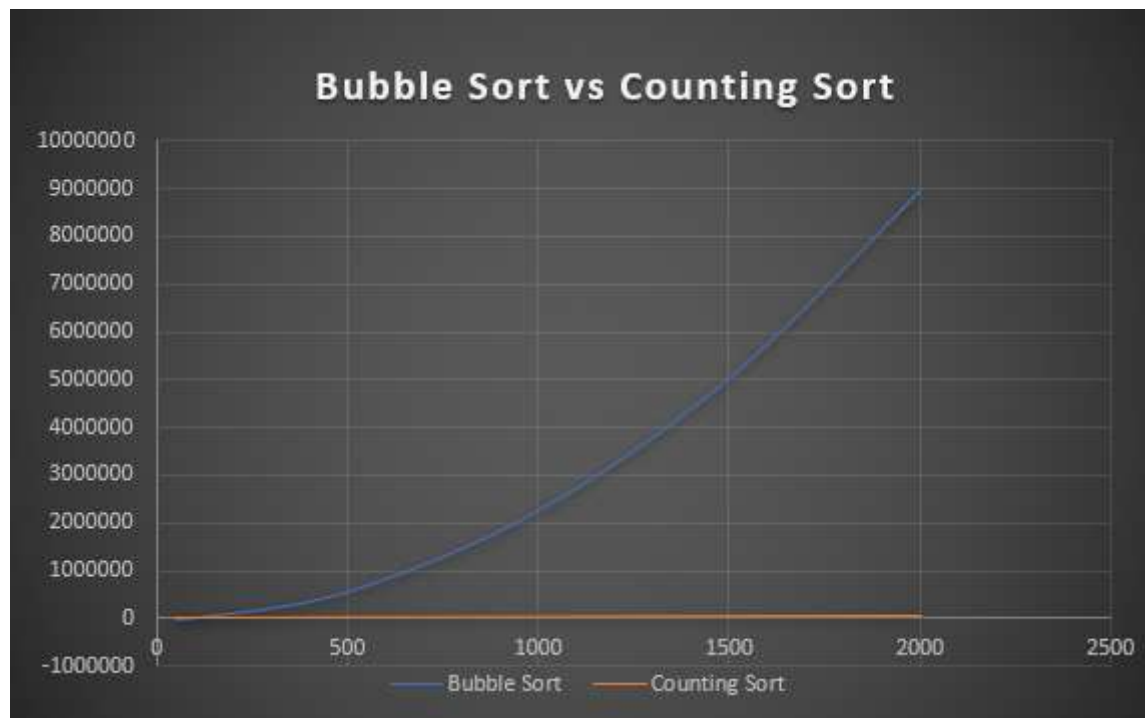
For 50 data :
Step count in Bubble Sort : 5561
Step count in Counting Sort : 60226
```

```
"F:\2-2\Study Materials\Sessional Based on CSE 2201\L6 Re...
Bubble sort :
35 103 188 288 382 778 1098 1104 111
2 1150 1322 1336 1536 1670 1722 1839 184
2 2051 2082 2188 2206 2286 2306 2315 235
3 2384 2436 2622 2646 2659 2702 2754 285
8 2927 3035 3289 3430 3548 3653 3803 393
0 3965 3976 3984 4031 4082 4219 4368 439
1 4624 4639 4765 4833 4944 4966 5005 502
1 5097 5349 5456 5537 5545 5572 5573 582
9 5889 6117 6270 6306 6511 6540 6729 677
5 6922 6940 6943 7348 7376 7444 7527 764
2 7751 8006 8587 8635 8723 8743 8755 890
9 8942 9040 9071 9161 9263 9628 9656 974
1 9758 9930 9953

Counting sort :
35 103 188 288 382 778 1098 1104 111
2 1150 1322 1336 1536 1670 1722 1839 184
2 2051 2082 2188 2206 2286 2306 2315 235
3 2384 2436 2622 2646 2659 2702 2754 285
8 2927 3035 3289 3430 3548 3653 3803 393
0 3965 3976 3984 4031 4082 4219 4368 439
1 4624 4639 4765 4833 4944 4966 5005 502
1 5097 5349 5456 5537 5545 5572 5573 582
9 5889 6117 6270 6306 6511 6540 6729 677
5 6922 6940 6943 7348 7376 7444 7527 764
2 7751 8006 8587 8635 8723 8743 8755 890
9 8942 9040 9071 9161 9263 9628 9656 974
1 9758 9930 9953

For 100 data :
Step count in Bubble Sort : 22425
Step count in Counting Sort : 60638
```

## Graph:





## Discussion:

In this problem we have seen a sorting algorithm of linear time complexity which is counting sort. Here we have also compared counting sort with bubble sort which is a comparison-based sorting algorithm. **Time complexity** of **bubble sort** is  $O(n^2)$  and **space complexity** is **1**. On the other hand **time complexity** of **counting sort** is  $O(n + k)$  and **space complexity** is  $O(n + k)$  where  $k$  is the range of inputs. We can also easily understand the complexity difference of both these algorithm from the graph. But we cannot use the counting sort for every sort because there is a **limitation of input range** and **space complexity**. For using counting sort, we must know the ranges of inputs and need a lot more space that's why we cannot use linear counting algorithms in every sorting operations.