Green University of Bangladesh
Department of Computer Science and Engineering(CSE)

Faculty of Sciences and Engineering

Semester: (Spring, Year:2021), B.Sc. in CSE (Day)

LAB REPORT NO 02

Course Title:Algorithms Lab

Course Code:206 Section:DB

Lab Experiment Name:

1. Create a text file containing 5000 or more numbers.
2. Apply Bubble sort, Selection sort and Insertion sort on those numbers.
3. Compare complexity of these three sorts on those 5000 or more numbers.

Student Details

	Name	ID
1.	Shamim Ahmed	201902067

Lab Date : 29/06/2021

Submission Date : 29/06/2021

Course Teacher's Name : Monoshi Kumar Roy

[For Teachers use only: Don't Write Anything inside this box]

L ab Report Status Marks: ……   **Signature:……**

.

1 TITLE OF THE LAB EXPERIMENT

1. Creating a text file containing 5000 numbers.
2. Applying Bubble sort, Selection sort and Insertion sort on those numbers.
3. Comparing complexity of these three sorts on those 5000 or more numbers.

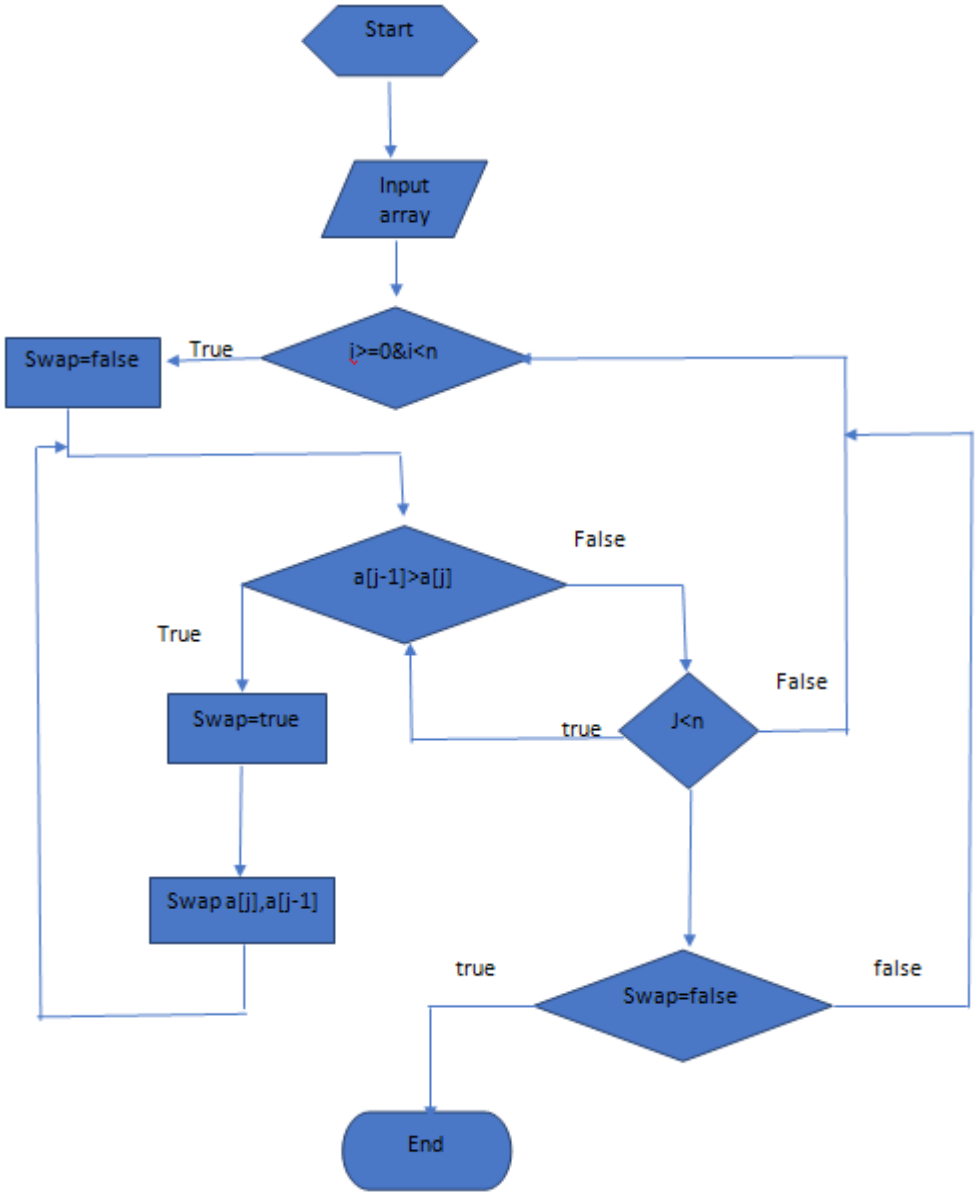
2 OBJECTIVES/AIM [1]

From this lab we will have a clear knowledge about shorting (Bubble sort, Selection sort and Insertion sort). We can implement those storing algorithm. We can compare them with having same complexity which one is better to work with.

3 ANALYSIS / DESIGN [2]

Bubble sort: Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order.

Bubble sort
Flow chart:



Bubble sort Algorithm:

Step 1: Start

Step 2: Read the array of given items from the user file.

Step 3: Take the first element(index = 0), compare the current element with the next element.

Step 4: If the current element is greater than the next element, swap them.

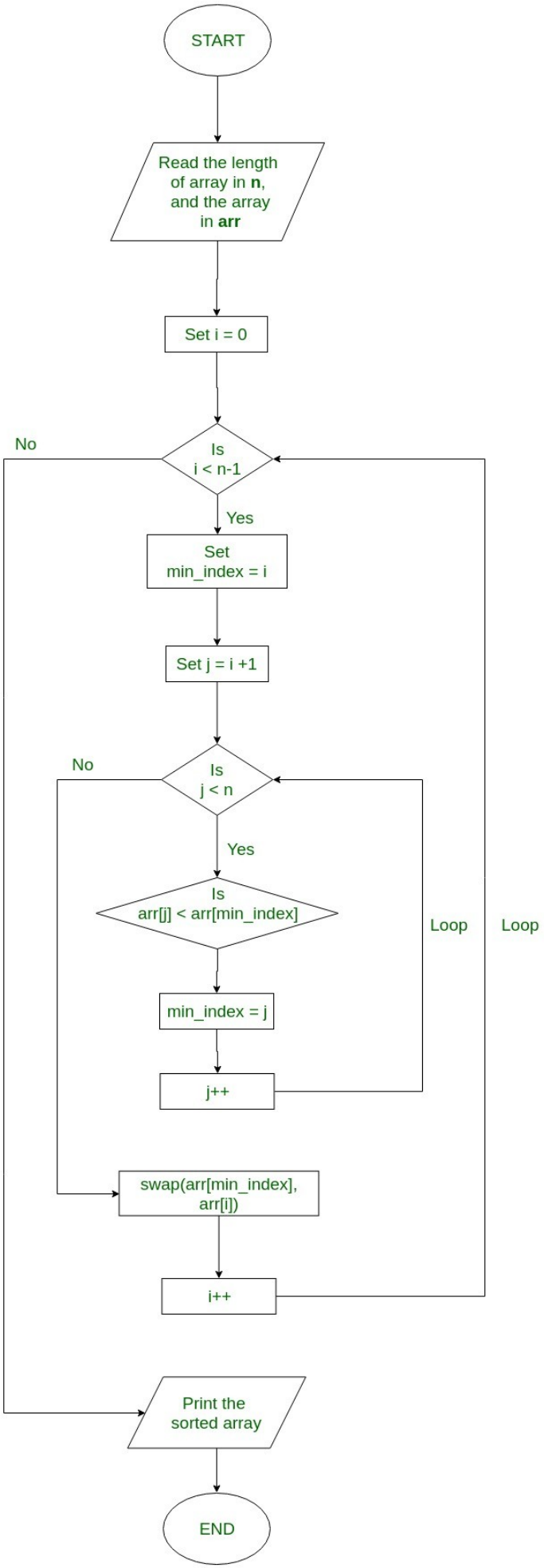
Step 5: Else,
If the current element is less than the next element, then move to the next element.

Step 6: Repeat Step 3 to Step 5 until all elements are sorted.

Step 7: Stop

Selection sort : The selection sort algorithm sorts an array by repeatedly finding the minimum element (considering ascending order) from unsorted part and putting it at the beginning. The algorithm maintains two subarrays in a given array.

Selection sort Flow chart:



Flowchart for Selection Sort

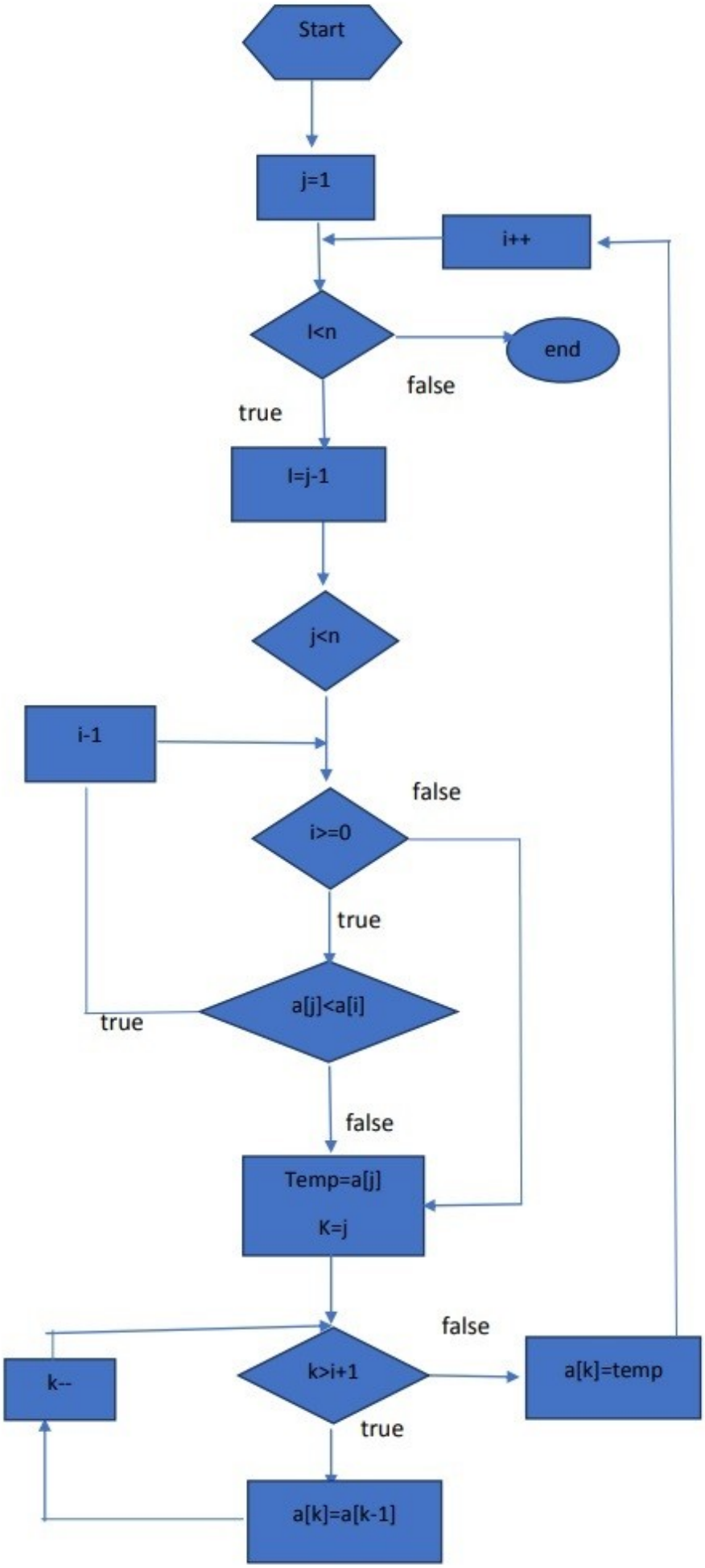
Selection sort Algorithm:

Let ARR is an array having N elements

1. Read ARR
2. Repeat step 3 to 6 for I=0 to N-1
3. Set MIN=ARR[I] and Set LOC=I
4. Repeat step 5 for J=I+1 to N
5. If MIN>ARR[J], then
   (a) Set MIN=ARR[J]
   (b) Set LOC=J
   [End of if]
   [End of step 4 loop]
6. Interchange ARR[I] and ARR[LOC] using temporary variable
   [End of step 2 outer loop]
7. Exit.

Insertion sort: Insertion sort is a simple sorting algorithm that works similar to the way we sort playing cards in your hands. The array is virtually split into a sorted and an unsorted part. Values from the unsorted part are picked and placed at the correct position in the sorted part.

Insertion sort Flow chart:


Start
j=1
i++
I<n
end
false
true
I=j-1
j<n
i-1
false
i>=0
true
a[j]<a[i]
true
false
Temp=a[j]
K=j
false
k--
k>i+1
a[k]=temp
true
a[k]=a[k-1]

Insertion sort Algorithm:

Let ARR is an array with N elements

1. Read ARR
2. Repeat step 3 to 8 for I=1 to N-1
3. Set Temp=ARR[I]
4. Set J=I-1
5. Repeat step 6 and 7 while Temp<ARR[J] AND J>=0

6. Set ARR[J+1]=ARR[J] [Moves element forward]
7. Set J=J-1
   [End of step 5 inner loop]
8. Set ARR[J+1]=Temp [Insert element in proper place]
   [End of step 2 outer loop]
9. Exit.

4 IMPLEMENTATION / PROCEDURE [2]

Bubble sort:

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

int main()
{
    ifstream input;
    ofstream output;
    int n=5000;
    int a[5000];

    input.open("Output.txt");
    output.open("Output_for_Selection.txt");
    int i,j, couter=0;

    for(int i=0;i<5000;i++)
    {
        input>>a[i];
    }

    for( i=0;i<n-1;i++)
    {
        for(j=0 ;j<n-i-1;j++)
        {
            if(a[j]>a[j+1])
            {
                swap(a[j],a[j+1]);
                //couter++;
            }
```

```
        }
    }
    for( i=0;i<n;i++)
    {
        cout<<a[i]<<" ";
        output<<a[i]<<" ";
    }
   // cout<<couter<<endl;
}
```

Selection sort :

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

int main()
{
    ifstream input;
    ofstream output;
    int n=5000;
    int a[5000];

    input.open("Output.txt");
    output.open("Output_for_Selection.txt");
    int i,j,counter=0;

    for(i=0;i<5000;i++)
    {
        input>>a[i];
    }

    for( i=0;i<n-1;i++)
    {
        for( j=i+1;j<n;j++)
        {
            if(a[i]>a[j])
            {
                swap(a[i],a[j]);
                //counter++;
            }
        }
    }
    for(int i=0;i<n;i++)
    {
```

```
        cout<<a[i]<<" ";
        output<<a[i]<<" ";
    }
    // cout<<couter<<endl;
}
```

Insertion sort:

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

int main()
{
    ifstream input;
    ofstream output;
    int n=5000;
    int a[5000];
    int counter=0;

    input.open("Output.txt");
    output.open("Output_for_Selection.txt");
    int i,j,temp;
    for(int i=0;i<5000;i++)
    {
        input>>a[i];
    }
    for(i=0;i<n-1;i++)
        {
            temp=a[i];
                j=i-1;

            while((temp<a[j])&&(j>=0))
            {
                a[j+1]=a[j];
                j=j-1;

            }

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

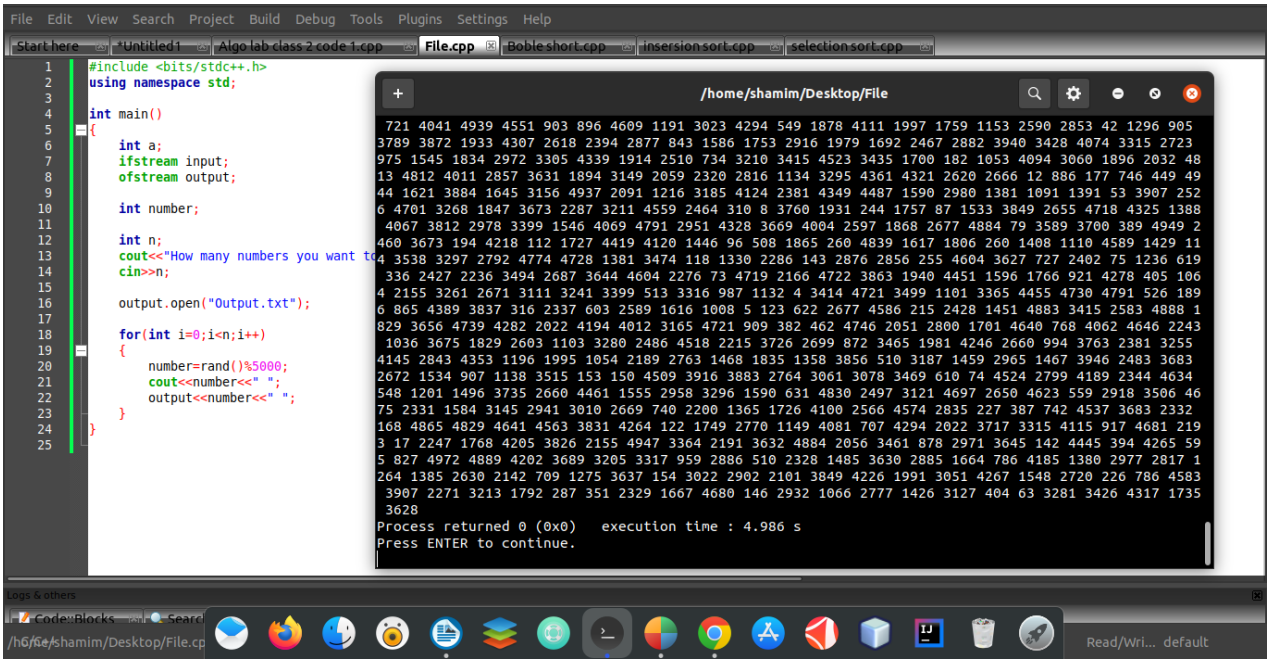
```
    for(int i=0;i<n;i++){
      cout<<a[i]<<" ";
       output<<a[i]<<" ";
    }
    //cout<<counter;
}
```

File:

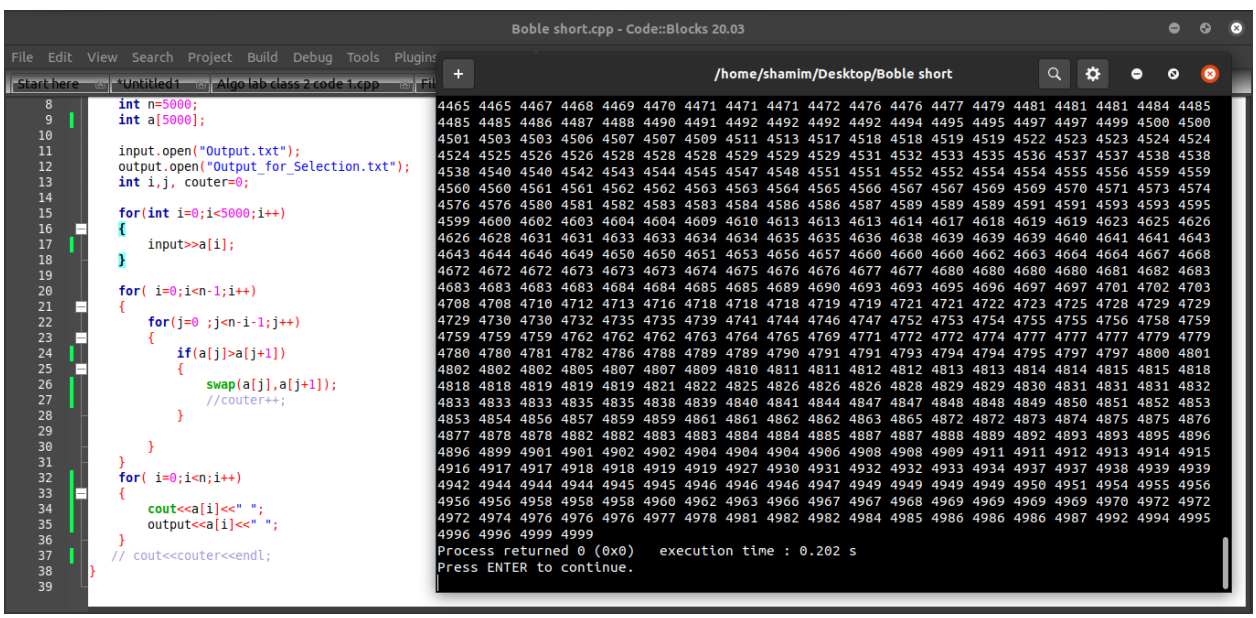
```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int a;
    ifstream input;
    ofstream output;

    int number;

    int n;
    cout<<"How many numbers you want to generate?:"<<endl;
    cin>>n;

    output.open("Output.txt");

    for(int i=0;i<n;i++)
    {
        number=rand()%5000;
        cout<<number<<" ";
        output<<number<<" ";
    }
}
```

5 TEST RESULT / OUTPUT [2]

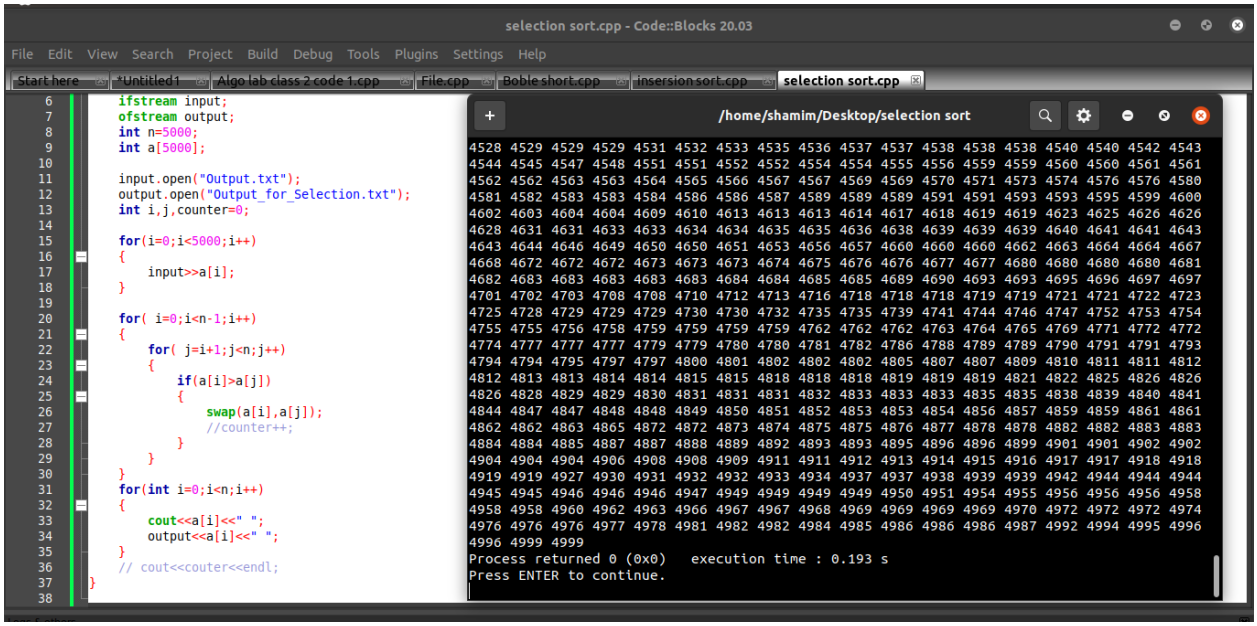
File:


File Edit View Search Project Build Debug Tools Plugins Settings Help
Start here
*Untitled1
Algo lab class 2 code 1.cpp
File.cpp
Boble short.cpp
insersion sort.cpp
selection sort.cpp
#include <bits/stdc++.h>
using namespace std;
int main()
{
int a;
ifstream input;
ofstream output;
int number;
int n;
cout<<"How many numbers you want to
cin>>n;
output.open("Output.txt");
for(int i=0;i<n;i++)
{
number=rand()%5000;
cout<<number<<" ";
output<<number<<" ";
}
}
/home/shamim/Desktop/File
721 4041 4939 4551 903 896 4609 1191 3023 4294 549 1878 4111 1997 1759 1153 2590 2853 42 1296 905
3789 3872 1933 4307 2618 2394 2877 843 1586 1753 2916 1979 1692 2467 2882 3940 3428 4074 3315 2723
975 1545 1834 2972 3305 4339 1914 2510 734 3210 3415 4523 3435 1700 182 1053 4094 3060 1896 2032 48
13 4812 4011 2857 3631 1894 3149 2059 2320 2816 1134 3295 4361 4321 2620 2666 12 886 177 746 449 49
44 1621 3884 1645 3156 4937 2091 1216 3185 4124 2381 4349 4487 1590 2980 1381 1091 1391 53 3907 252
6 4701 3268 1847 3673 2287 3211 4559 2464 310 8 3760 1931 244 1757 87 1533 3849 2655 4718 4325 1388
4067 3812 2978 3399 1546 4069 4791 2951 4328 3669 4004 2597 1868 2677 4884 79 3589 3700 389 4949 2
460 3673 194 4218 112 1727 4419 4120 1446 96 508 1865 260 4839 1617 1806 260 1408 1110 4589 1429 11
4 3538 3297 2792 4774 4728 1381 3474 118 1330 2286 143 2876 2856 255 4604 3627 727 2402 75 1236 619
336 2427 2236 3494 2687 3644 4604 2276 73 4719 2166 4722 3863 1940 4451 1596 1766 921 4278 405 106
4 2155 3261 2671 3111 3241 3399 513 3316 987 1132 4 3414 4721 3499 1101 3365 4455 4730 4791 526 189
6 865 4389 3837 316 2337 603 2589 1616 1008 5 123 622 2677 4586 215 2428 1451 4883 3415 2583 4888 1
829 3656 4739 4282 2022 4194 4012 3165 4721 909 382 462 4746 2051 2800 1701 4640 768 4062 4646 2243
1036 3675 1829 2603 1103 3280 2486 4518 2215 3726 2699 872 3465 1981 4246 2660 994 3763 2381 3255
4145 2843 4353 1196 1995 1054 2189 2763 1468 1835 1358 3856 510 3187 1459 2965 1467 3946 2483 3683
2672 1534 907 1138 3515 153 150 4509 3916 3883 2764 3061 3078 3469 610 74 4524 2799 4189 2344 4634
548 1201 1496 3735 2660 4461 1555 2958 3296 1590 631 4830 2497 3121 4697 2650 4623 559 2918 3506 46
75 2331 1584 3145 2941 3010 2669 740 2200 1365 1726 4100 2566 4574 2835 227 387 742 4537 3683 2332
168 4865 4829 4641 4563 3831 4264 122 1749 2770 1149 4081 707 4294 2022 3717 3315 4115 917 4681 219
3 17 2247 1768 4205 3826 2155 4947 3364 2191 3632 4884 2056 3461 878 2971 3645 142 4445 394 4265 59
5 827 4972 4889 4202 3689 3205 3317 959 2886 510 2328 1485 3630 2885 1664 786 4185 1380 2977 2817 1
264 1385 2630 2142 709 1275 3637 154 3022 2902 2101 3849 4226 1991 3051 4267 1548 2720 226 786 4583
3907 2271 3213 1792 287 351 2329 1667 4680 146 2932 1066 2777 1426 3127 404 63 3281 3426 4317 1735
3628
Process returned 0 (0x0) execution time : 4.986 s
Press ENTER to continue.
Logs & others
Code::Blocks
/home/shamim/Desktop/File.cp
Read/Wri... default

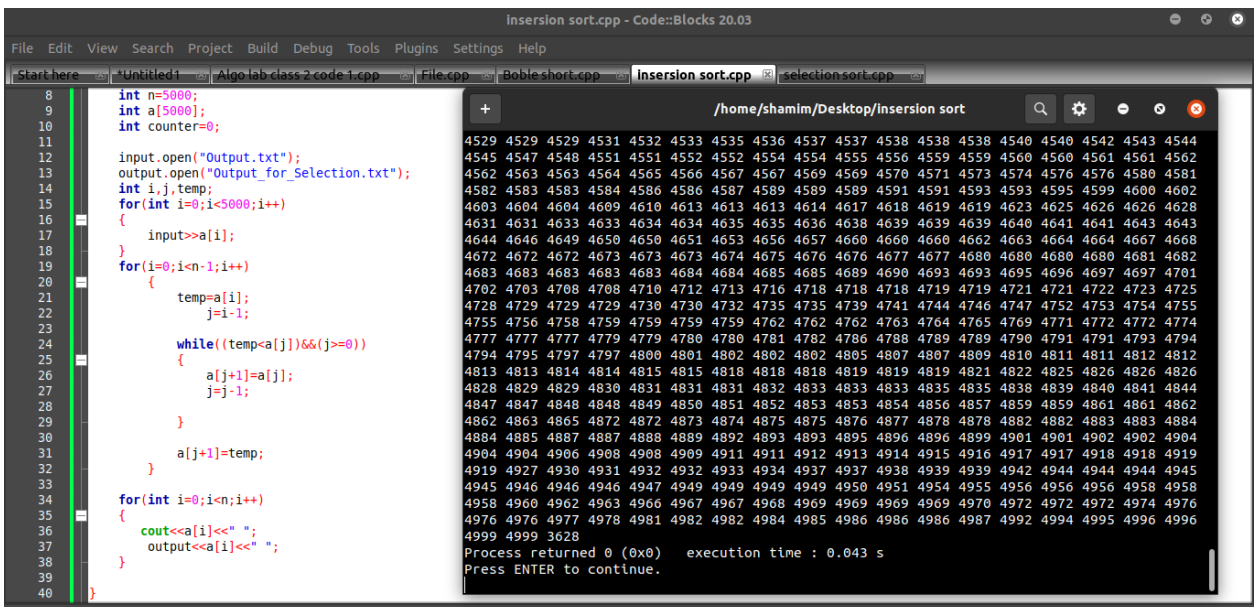
Bubble sort:


Boble short.cpp - Code::Blocks 20.03
File Edit View Search Project Build Debug Tools Plugins
Start here
*Untitled1
Algo lab class 2 code 1.cpp
int n=5000;
int a[5000];
input.open("Output.txt");
output.open("Output_for_Selection.txt");
int i,j, couter=0;
for(int i=0;i<5000;i++)
{
input>>a[i];
}
for(i=0;i<n-1;i++)
{
for(j=0 ;j<n-i-1;j++)
{
if(a[j]>a[j+1])
{
swap(a[j],a[j+1]);
//couter++;
}
}
}
for(i=0;i<n;i++)
{
cout<<a[i]<<" ";
output<<a[i]<<" ";
}
// cout<<couter<<endl;
}
/home/shamim/Desktop/Boble short
4465 4465 4467 4468 4469 4470 4471 4471 4471 4472 4476 4476 4477 4479 4481 4481 4481 4484 4485
4485 4485 4486 4487 4488 4490 4491 4492 4492 4492 4492 4494 4495 4495 4497 4497 4499 4500 4500
4501 4503 4503 4506 4507 4507 4509 4511 4513 4517 4518 4518 4519 4519 4522 4523 4523 4524 4524
4524 4525 4526 4526 4528 4528 4528 4529 4529 4529 4531 4532 4533 4535 4536 4537 4537 4538 4538
4538 4540 4540 4542 4543 4544 4545 4547 4548 4551 4551 4552 4552 4554 4554 4555 4556 4559 4559
4560 4560 4561 4561 4562 4562 4563 4563 4564 4565 4566 4567 4567 4569 4569 4570 4571 4573 4574
4576 4576 4580 4581 4582 4583 4583 4584 4586 4586 4587 4589 4589 4589 4591 4591 4593 4593 4595
4599 4600 4602 4603 4604 4604 4609 4610 4613 4613 4613 4614 4617 4618 4619 4619 4623 4625 4626
4626 4628 4631 4631 4633 4633 4634 4634 4635 4635 4636 4638 4639 4639 4639 4640 4641 4641 4643
4643 4644 4646 4649 4650 4650 4651 4653 4656 4657 4660 4660 4660 4662 4663 4664 4664 4667 4668
4672 4672 4672 4673 4673 4673 4674 4675 4676 4676 4677 4677 4680 4680 4680 4680 4681 4682 4683
4683 4683 4683 4683 4684 4684 4685 4685 4689 4690 4693 4693 4695 4696 4697 4697 4701 4702 4703
4708 4708 4710 4712 4713 4716 4718 4718 4718 4719 4719 4721 4721 4722 4723 4725 4728 4729 4729
4729 4730 4730 4732 4735 4735 4739 4741 4744 4746 4747 4752 4753 4754 4755 4755 4756 4758 4759
4759 4759 4759 4762 4762 4762 4763 4764 4765 4769 4771 4772 4772 4774 4777 4777 4777 4779 4779
4780 4780 4781 4782 4786 4788 4789 4789 4790 4791 4791 4793 4794 4794 4795 4797 4797 4800 4801
4802 4802 4802 4805 4807 4807 4809 4810 4811 4811 4812 4812 4813 4813 4814 4814 4815 4815 4818
4818 4818 4819 4819 4819 4821 4822 4825 4826 4826 4826 4828 4829 4829 4830 4831 4831 4831 4832
4833 4833 4833 4835 4835 4838 4839 4840 4841 4844 4847 4847 4848 4848 4849 4850 4851 4852 4853
4853 4854 4856 4857 4859 4859 4861 4861 4862 4862 4863 4865 4872 4872 4873 4874 4875 4875 4876
4877 4878 4878 4882 4882 4883 4883 4884 4884 4885 4887 4887 4888 4889 4892 4893 4893 4895 4896
4896 4899 4901 4901 4902 4902 4904 4904 4904 4906 4908 4908 4909 4911 4911 4912 4913 4914 4915
4916 4917 4917 4918 4918 4919 4919 4927 4930 4931 4932 4932 4933 4934 4937 4937 4938 4939 4939
4942 4944 4944 4944 4945 4945 4946 4946 4946 4947 4949 4949 4949 4949 4949 4950 4951 4954 4955 4956
4956 4956 4958 4958 4958 4960 4962 4963 4966 4967 4967 4968 4969 4969 4969 4969 4970 4972 4972
4972 4974 4976 4976 4976 4977 4978 4981 4982 4982 4984 4985 4986 4986 4986 4987 4992 4994 4995
4996 4996 4999 4999
Process returned 0 (0x0) execution time : 0.202 s
Press ENTER to continue.

Selection sort :



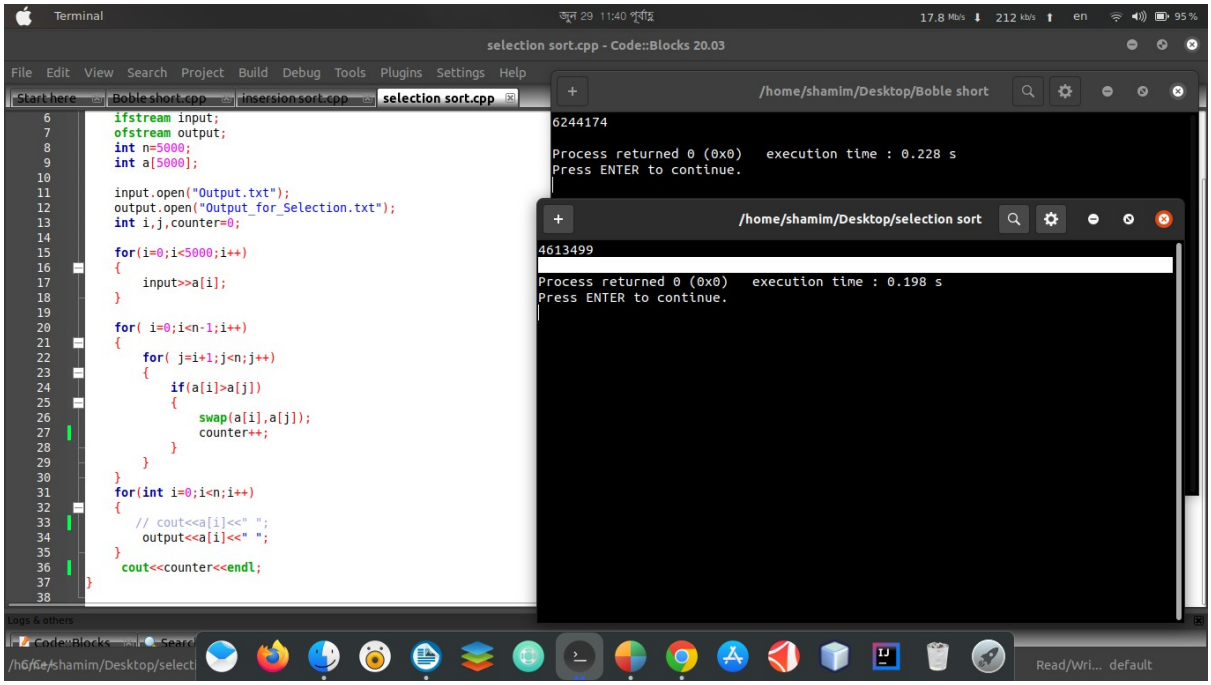
Insertion sort:



6 ANALYSIS AND DISCUSSION

I. Bubble sort, Selection sort and Insertion sort all have the same worst time complexity O[N2].

II. Though they have same worst time complexity when we use a counter on your code we saw in bubble sort the inner loop run for 6244174 times and where selection sort run for 4613599 for random 5000 numbers.



III. we can say that they have same worst time complexity but Insertion is best on them than Selection sort and worst them is Bubble sort.

IV. I was facing problem to declare file and take input from it. With my teachers help I over come from it.

V. I learn about those 3 sorting algorithm. Now I can compare them .