**LAB PRACTICALS (JAN-JUN, 2025)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.**  **No.** | **WEEK**  **No.** | **NAME OF PRACTICAL** | **Date** | **Signature** |
| **1.** | **WEEK 1** |  |  |  |
| **2.** | **WEEK 2** |  |  |  |
| **3.** | **WEEK 3** |  |  |  |
| **4.** | **WEEK 4** |  |  |  |
| **5.** | **WEEK 5** |  |  |  |
| **6.** | **WEEK 6** |  |  |  |
| **7.** | **WEEK 7** |  |  |  |
| **8.** | **WEEK 8** |  |  |  |
| **9.** | **WEEK 9** |  |  |  |
| **10.** | **WEEK**  **10** |  |  |  |
| **11.** | **WEEK 11** |  |  |  |
| **12.** | **WEEK**  **12** |  |  |  |
| **13.** | **WEEK**  **13** |  |  |  |
| **14.** | **WEEK**  **14** |  |  |  |

**LAB 01**

TASK 01:

**BUBBLE SORT**

CODE:

**#**include <bits/stdc++.h>

using namespace std;

void bubblesort(int arr[] , int n){

    int count=0;

    for(int i=0;i<n-1;i++){

        bool swapped = false;

        for(int j=0;j<n-i-1;j++){

            if(arr[j] > arr[j+1]){

                swap(arr[j],arr[j+1]);

                swapped = true;

            }

            count++;

        }

        if(!swapped){break;}

    }

    cout<<count<<endl;

}

void print(int arr[] ,int n){

    for(int i=0;i<n;i++){

        cout<<arr[i]<<" ";

    }

    cout<<endl;

}

int main(){

    int arr[7] = {7,6,5,4,3,2,1};

    int n = 7;

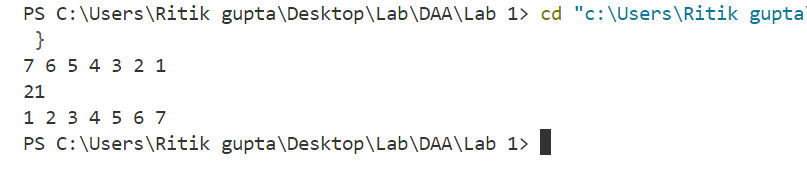
    print(arr,n);

    bubblesort(arr,n);

    print(arr,n);

}

**OUTPUT:**



**SELECTION SORT:**

#include <bits/stdc++.h>

using namespace std;

void selectionsort(int arr[] , int n){

int count =0;

for(int i =0;i<n-1;i++){

int min\_idx = i;

for(int j = i+1;j<n;j++){

if(arr[j] < arr[min\_idx]){

min\_idx = j;

}

count++;

}

swap(arr[min\_idx],arr[i]);

}

cout<<count<<endl;

}

void print(int arr[] ,int n){

for(int i=0;i<n;i++){

cout<<arr[i]<<" ";

}

cout<<endl;

}

int main(){

int arr[5] = {3,8,2,5,1};

int n = 5;

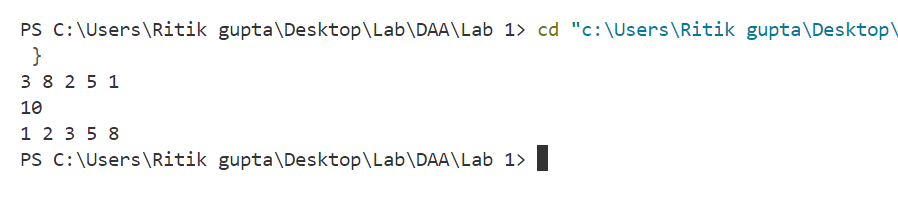
print(arr,n);

selectionsort(arr,n);

print(arr,n);

}

**OUTPUT:**

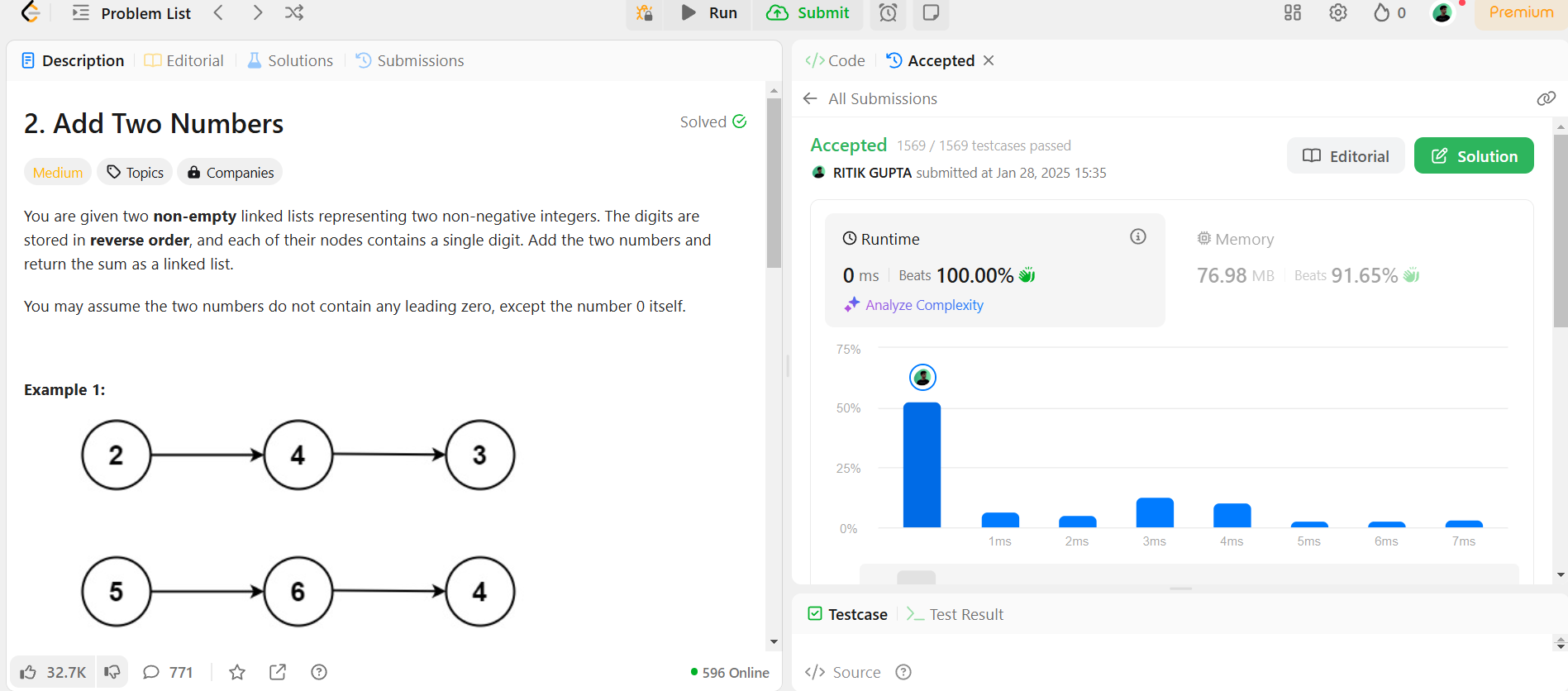
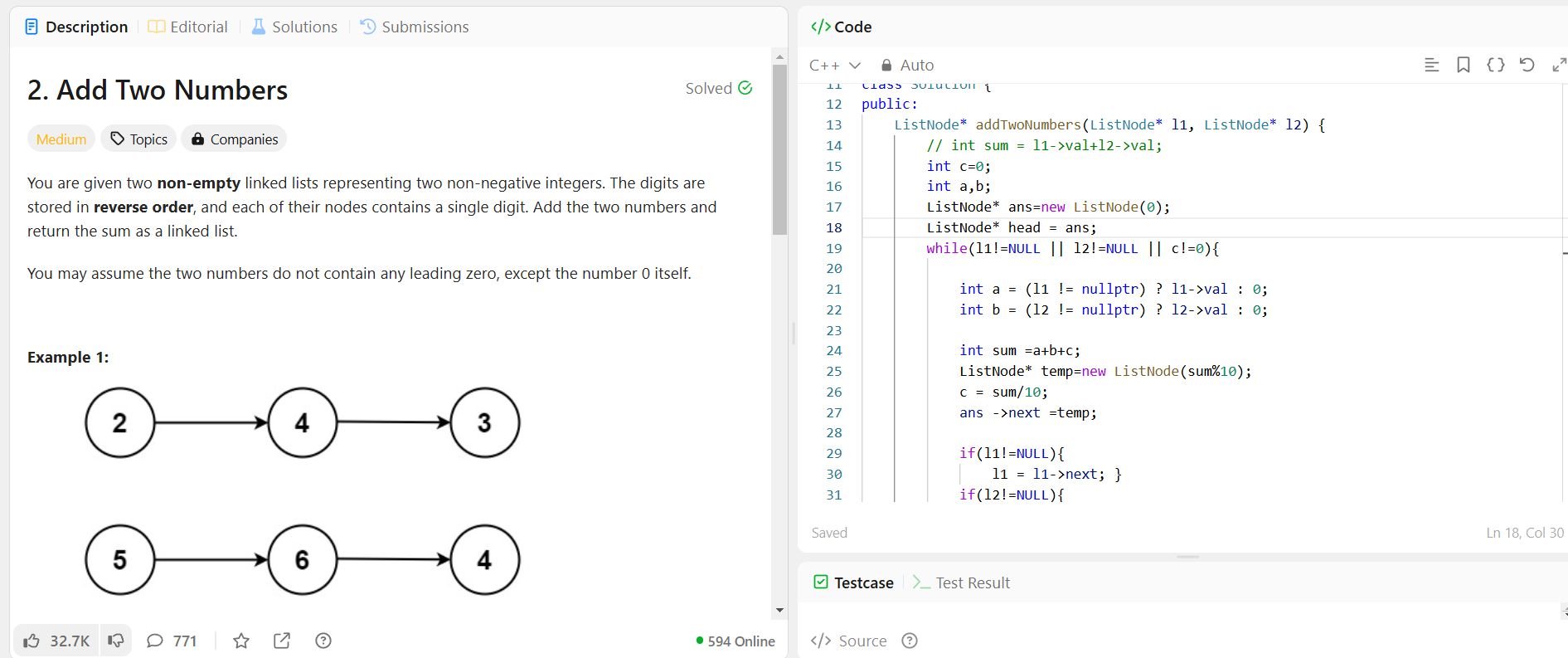


**Time Complexities:**

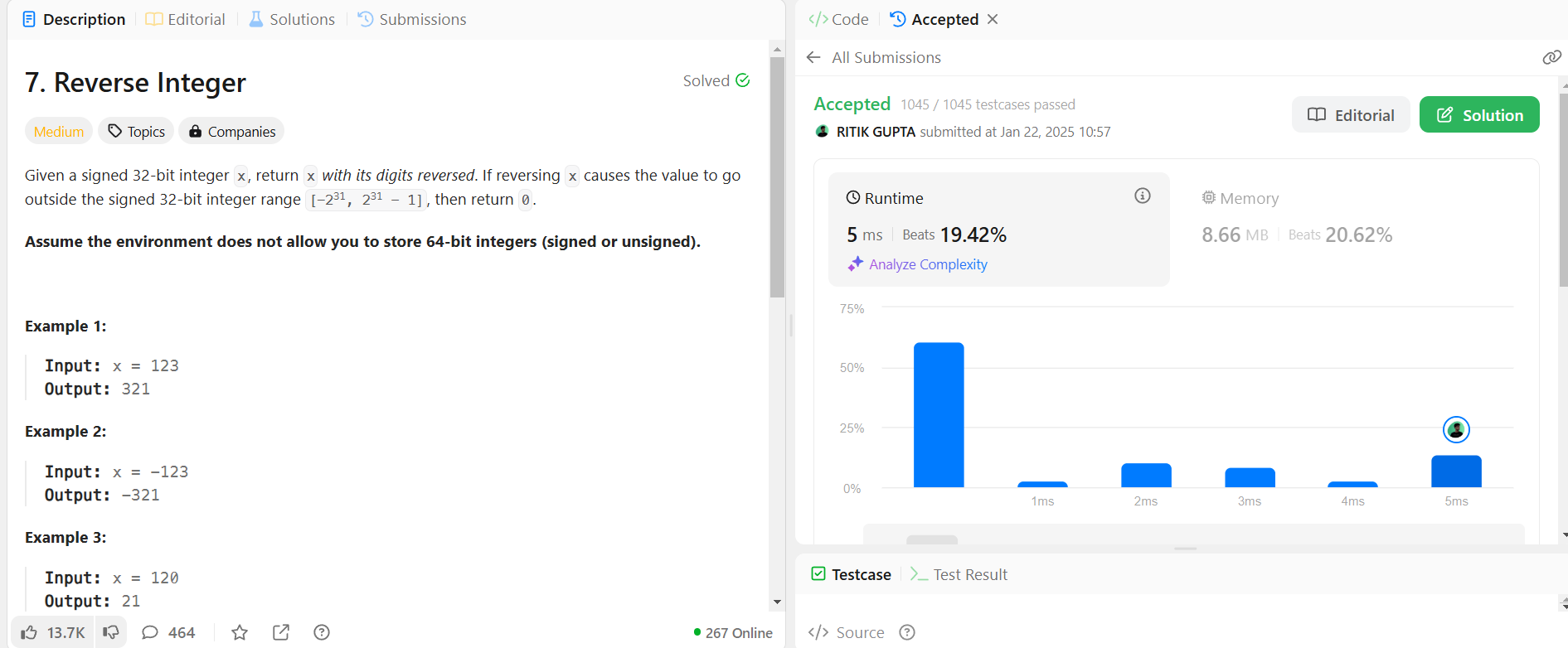
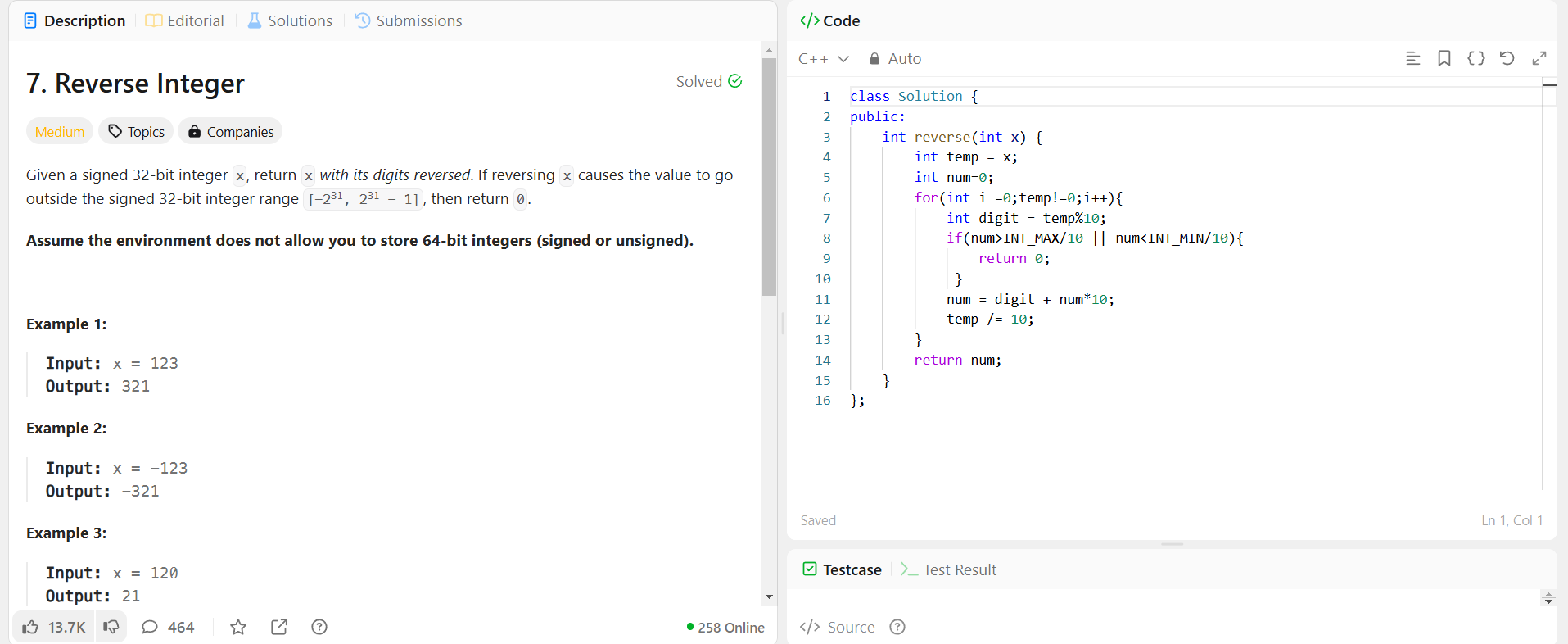
|  |  |  |
| --- | --- | --- |
|  | **Bubble sort** | **Selection sort** |
| **Best case** | **n.(n-1)/2** | **n.(n-1)/2** |
| **Worst case** | **n-1** | **n.(n-1)/2** |

TASK-02:

1**.**Add two numbers

****

2.Reverse integer

****

3.Dutch flag

