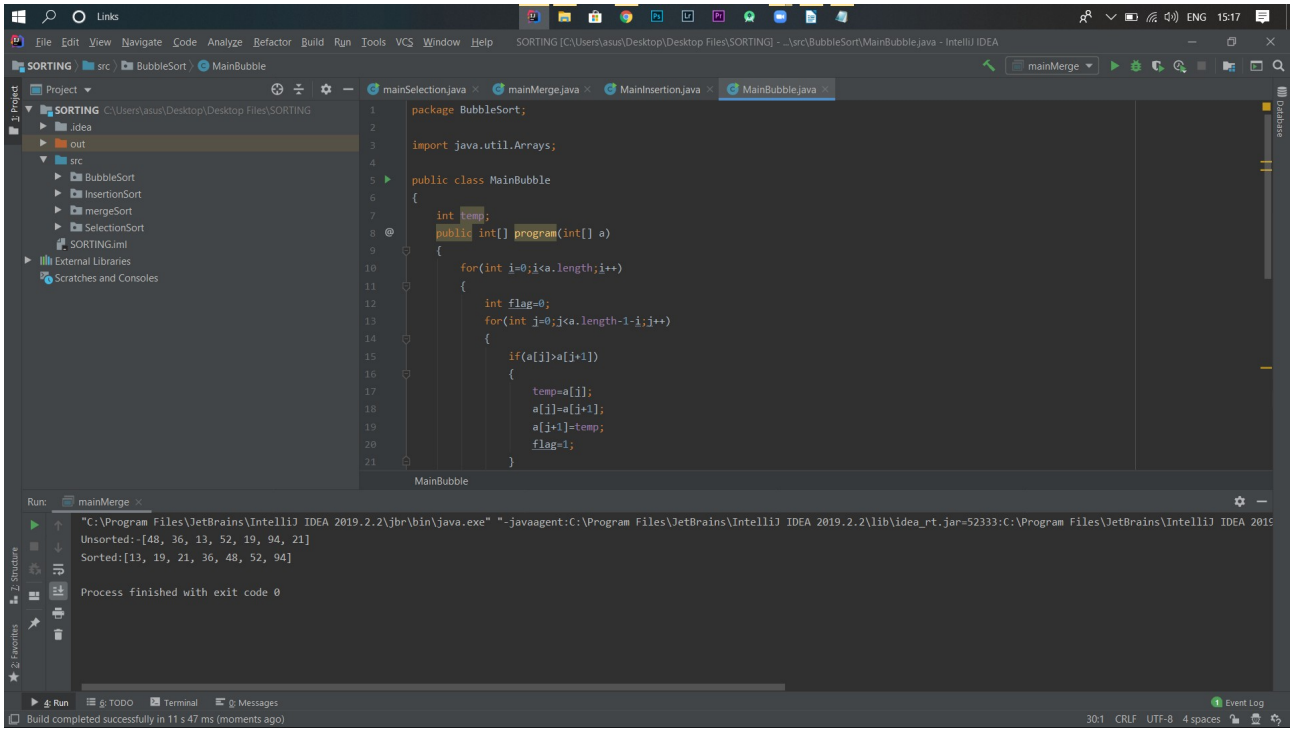
BUBBLE SORT CODE:-

```
package BubbleSort;

import java.util.Arrays;

public class MainBubble
{
    int temp;
    public int[] program(int[] a)
    {
        for(int i=0;i<a.length;i++)
        {
            int flag=0;
            for(int j=0;j<a.length-1-i;j++)
            {
                if(a[j]>a[j+1])
                {
                    temp=a[j];
                    a[j]=a[j+1];
                    a[j+1]=temp;
                    flag=1;
                }
            }
            if(flag==0)
            {
                break;
            }
        }
        return a;
    }

    public static void main(String[] args)
    {
        int a[]={36,19,29,12,5};
        System.out.println("Unsorted:-"+ Arrays.toString(a));
        MainBubble obj=new MainBubble();
        obj.program(a);
        System.out.println(Arrays.toString(a));
    }
}
```

INPUTS:- {36,19,29,12,5}

Output:-Sorted:[13, 19, 21, 36, 48, 52, 94]



COMPLEXITY:-

Worst and Average Case Time Complexity : O(n*n). Worst case occurs when array is reverse sorted.

Best Case Time Complexity: O(n). Best case occurs when array is already sorted.

ALGORITHM:

Given a list L of n elements with values or records L0, L1, …, Ln-1, bubble sort is applied to sort the list L.

1. Compare first two elements L0, L1 in the list.

2. if L1 < L0, swap those elements and continue with next 2 elements.

3. Repeat the same step until whole the list is sorted, so no more swaps are possible.

4. Return the final sorted list.