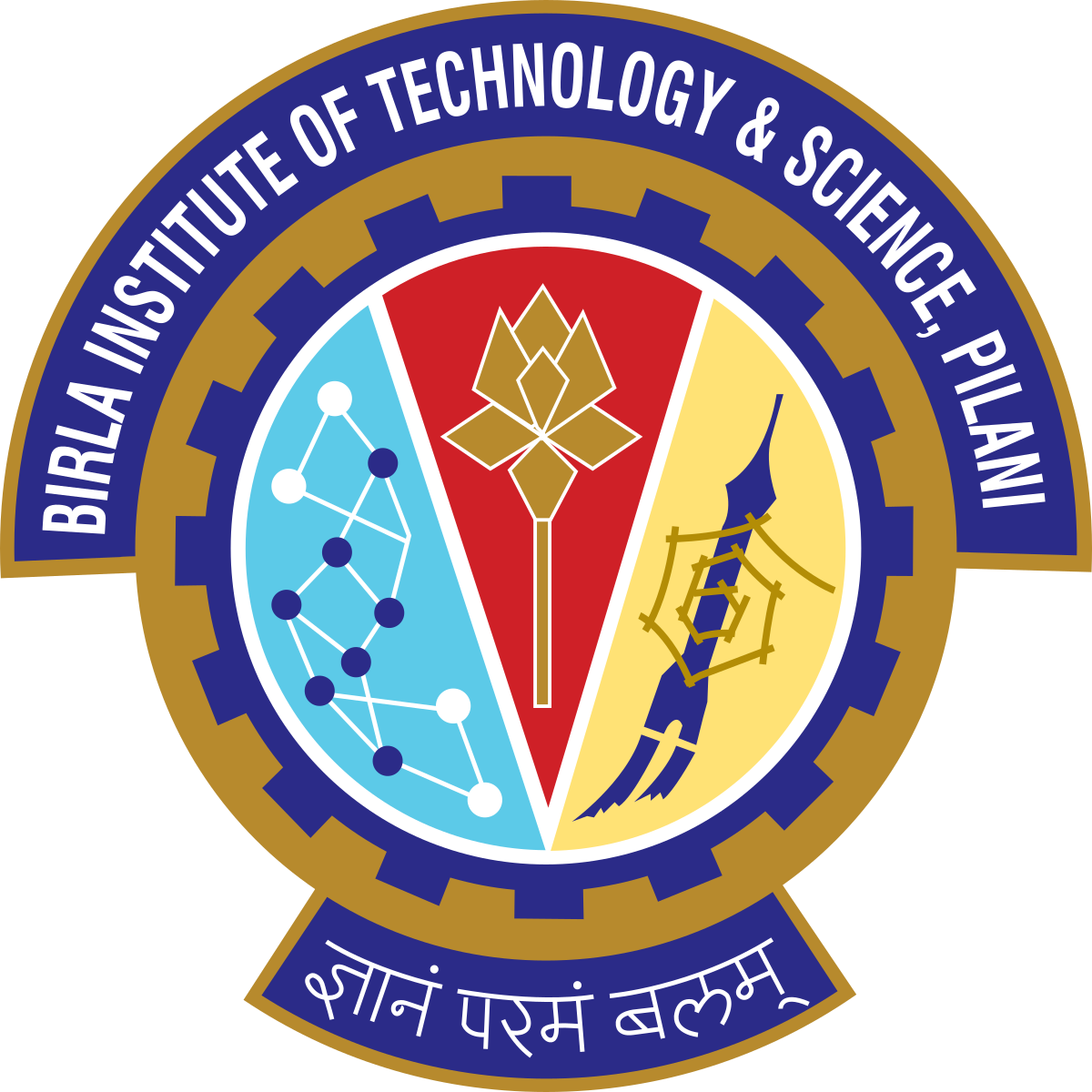
**ANALYSIS REPORT**

**(Time Analysis)**

****

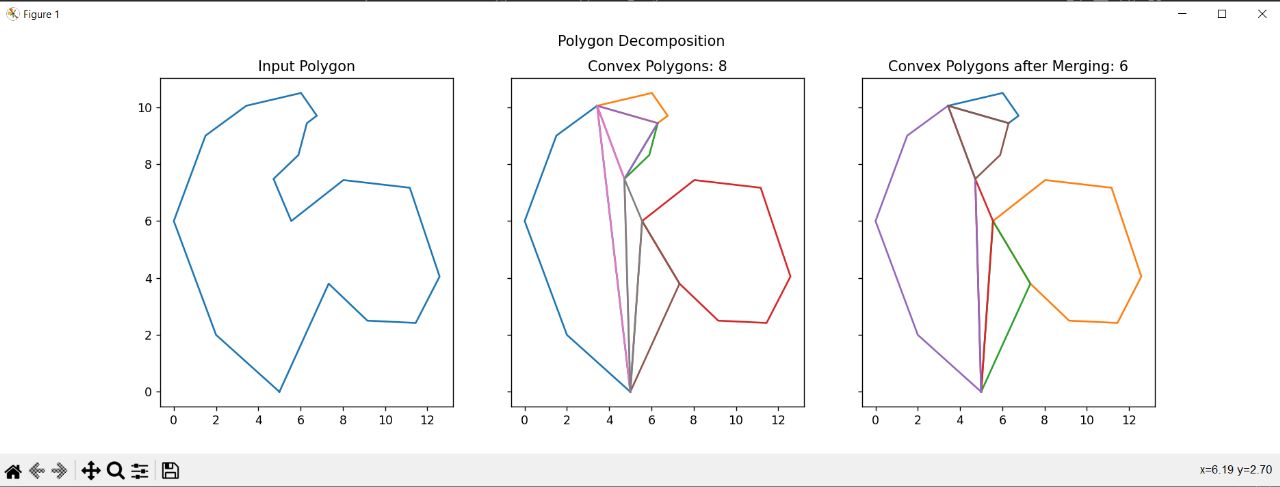
**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI**

**HYDERABAD CAMPUS**

**Authors:**

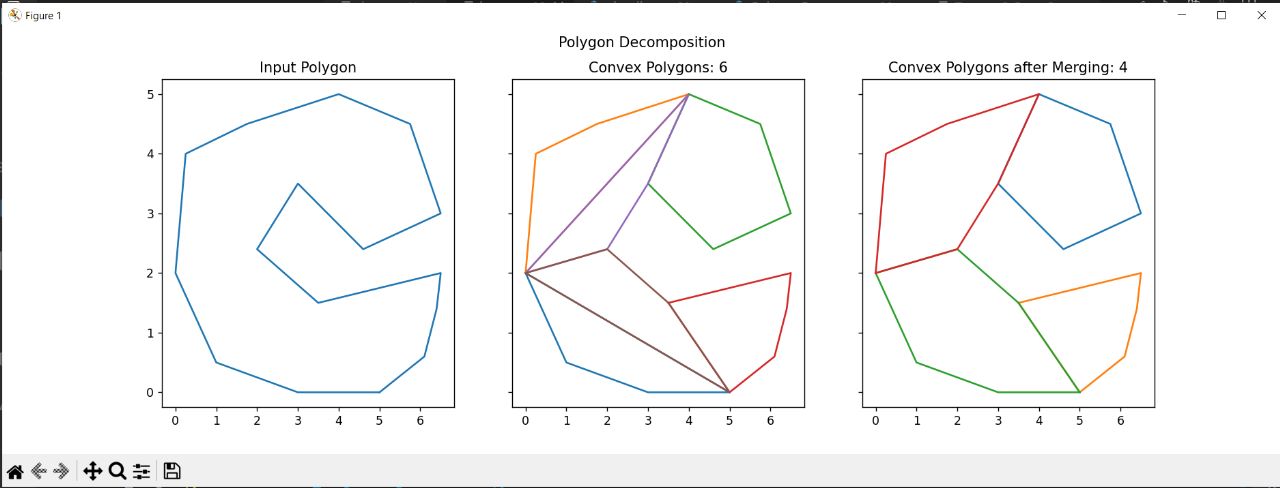
1. Manasa SK 2020A7PS0223H
2. Ayush Agarwal 2020A7PS0160H
3. Vedant Mathur 2020A7PS2047H
4. Dhairya Agrawal 2020A7PS0130H

**TestCase 1 (Paper):**



Algorithm Running Time: **6.64e-05**

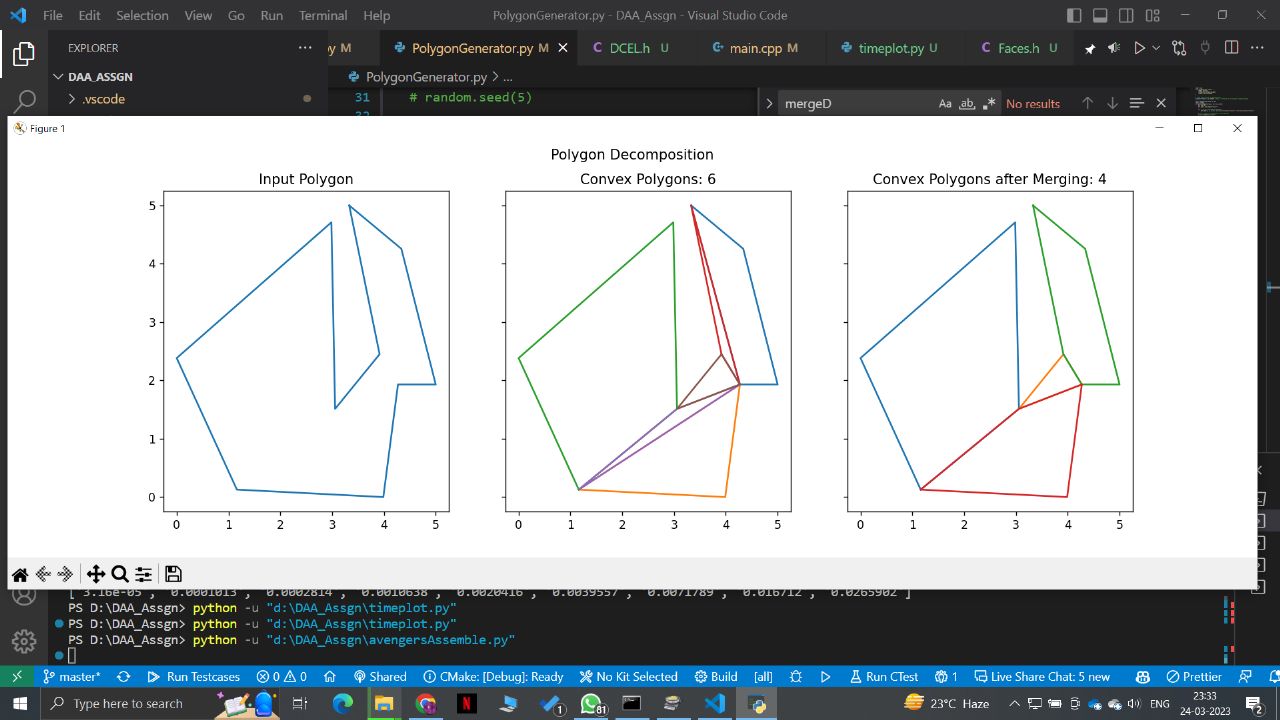
Merging Running Time: **7.6e-06**

**TestCase 2 (Paper):**

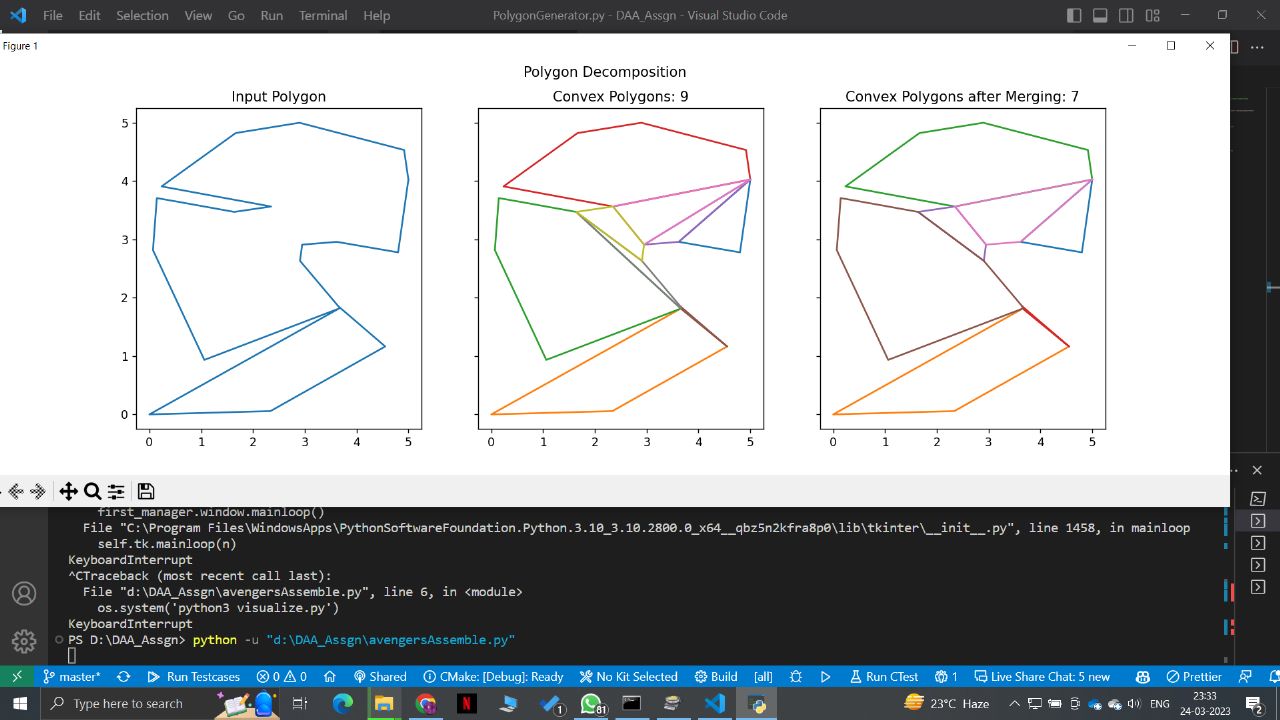
Algorithm Running Time: **4.44e-05**

Merging Running Time: **6.7e-06**

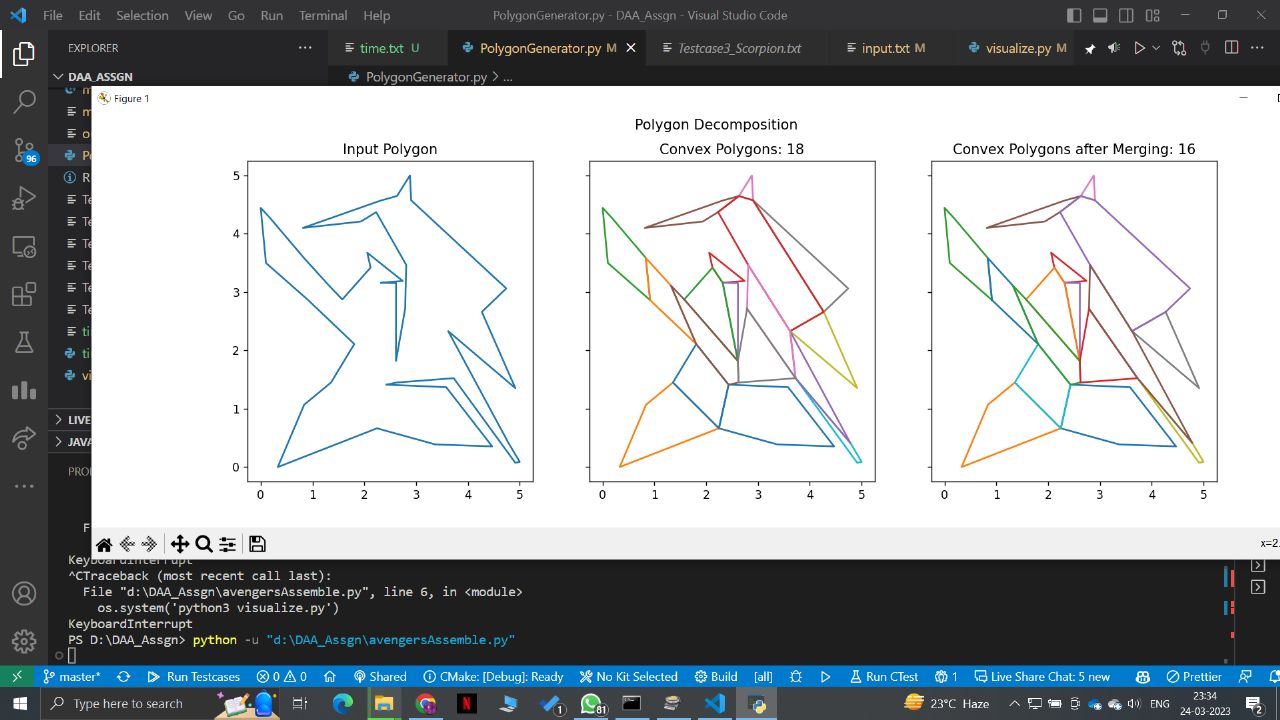
**TestCase 3:**

Number of Points: 10

**TestCase 4:**

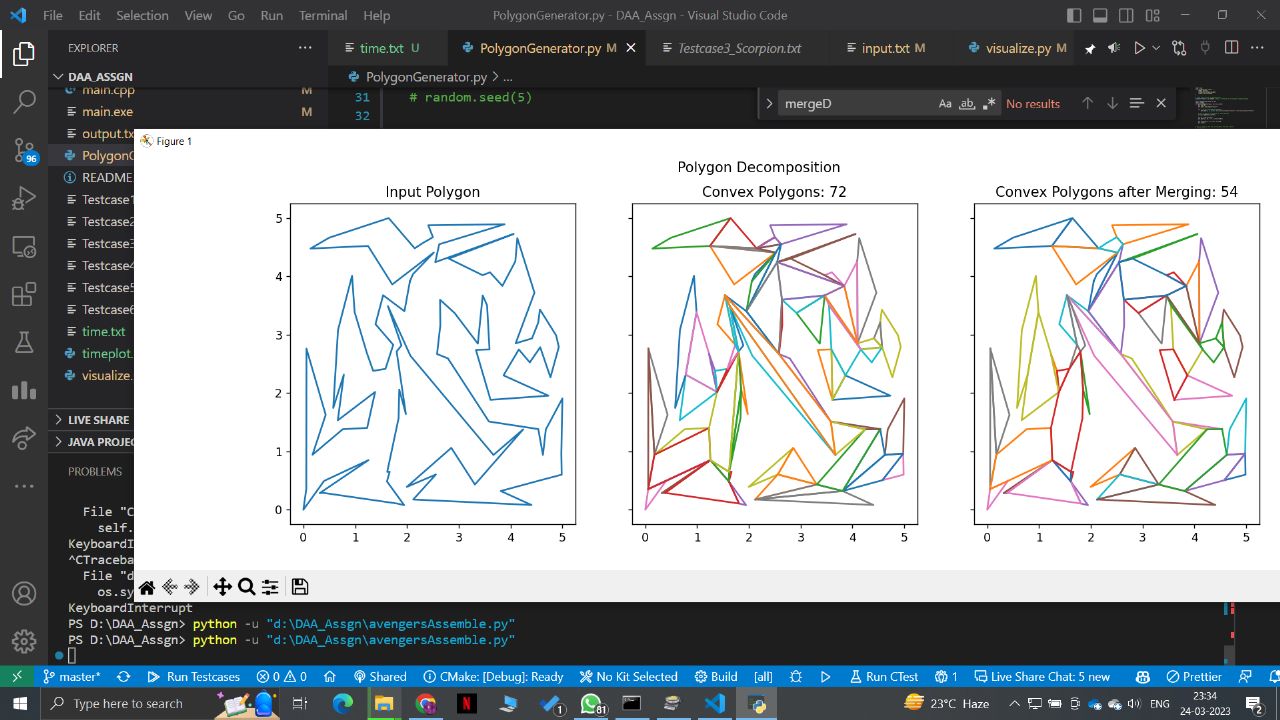
Number of Points: 20

**TestCase 5:**



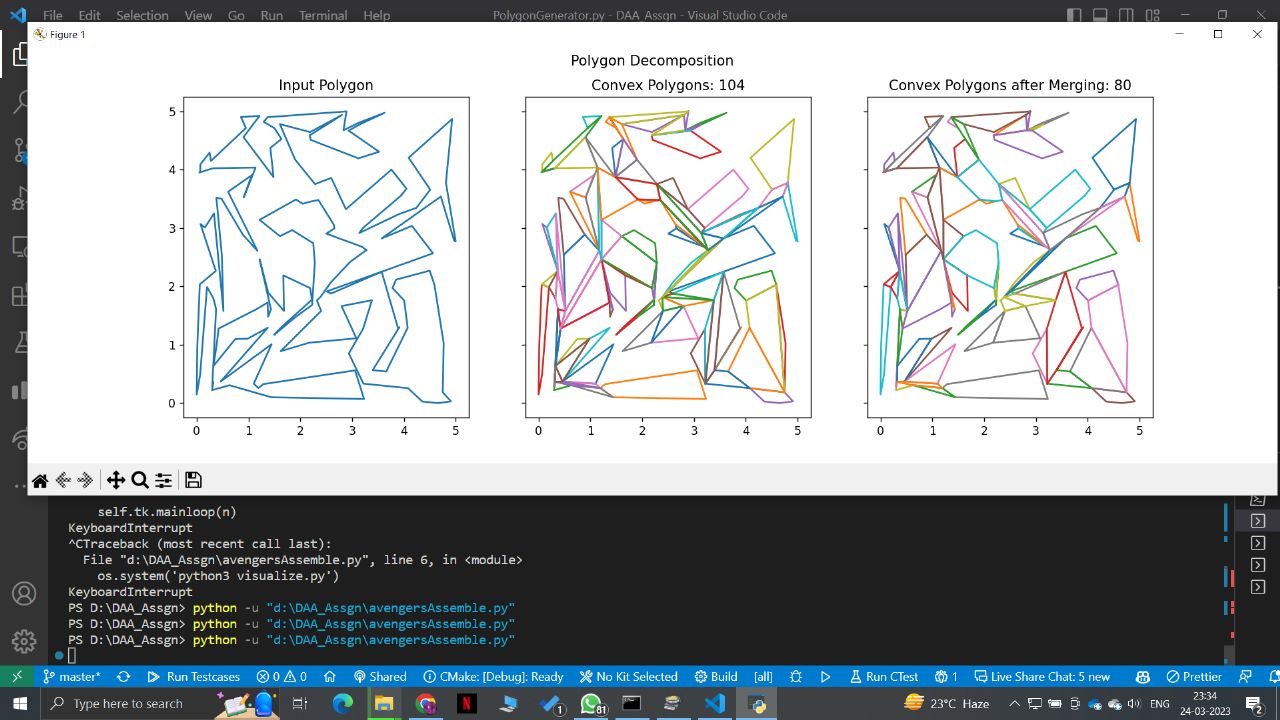
Number of Points:40

**TestCase 6:**

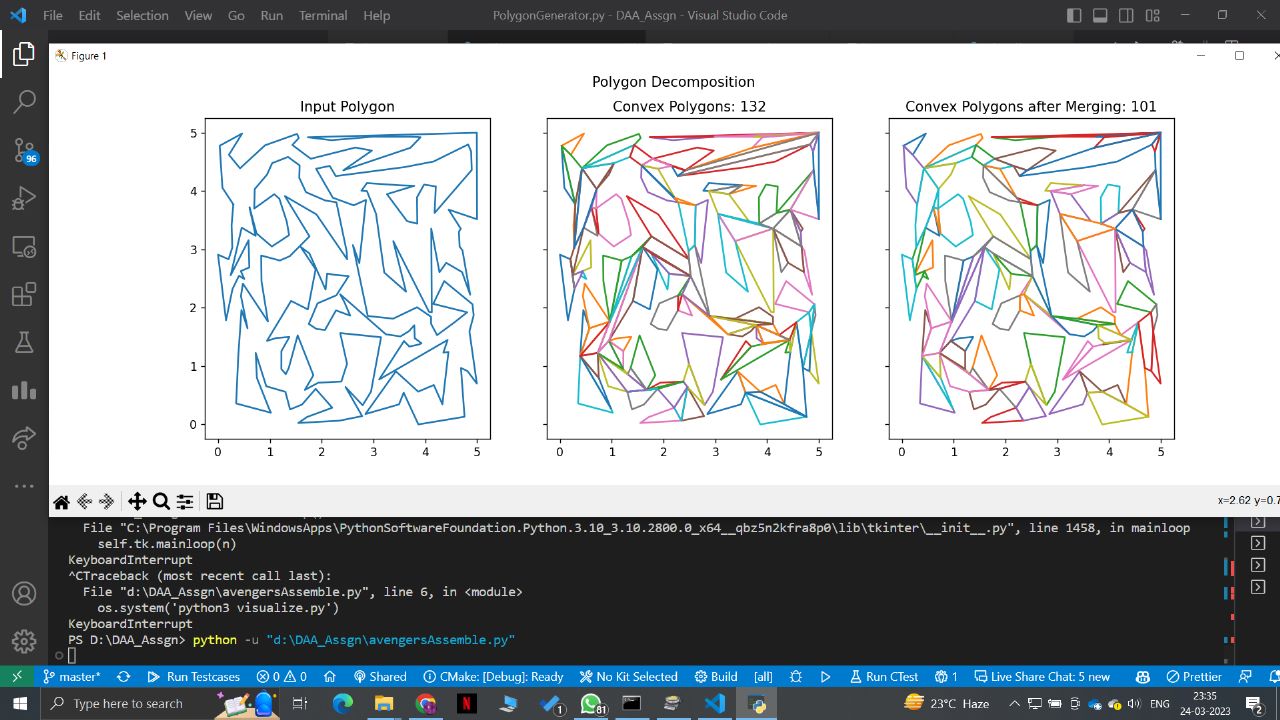


Number of Points:100

**TestCase 7:**

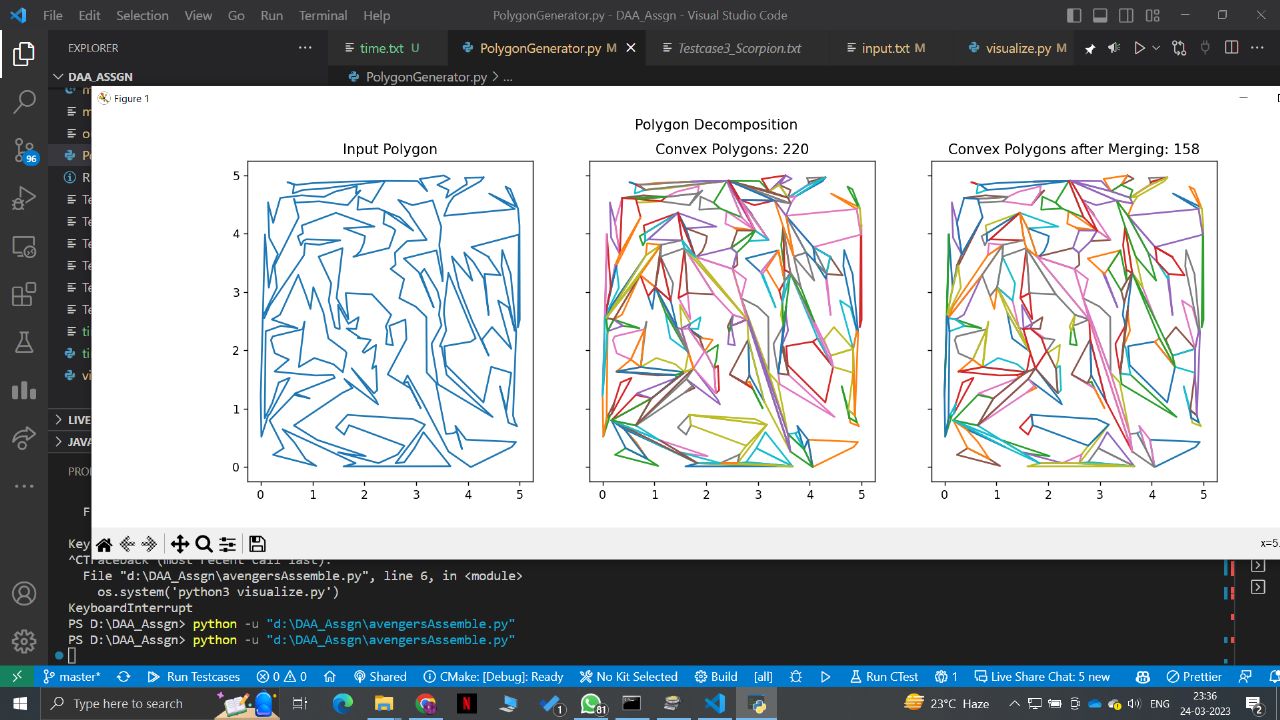
****

Number of Points:150

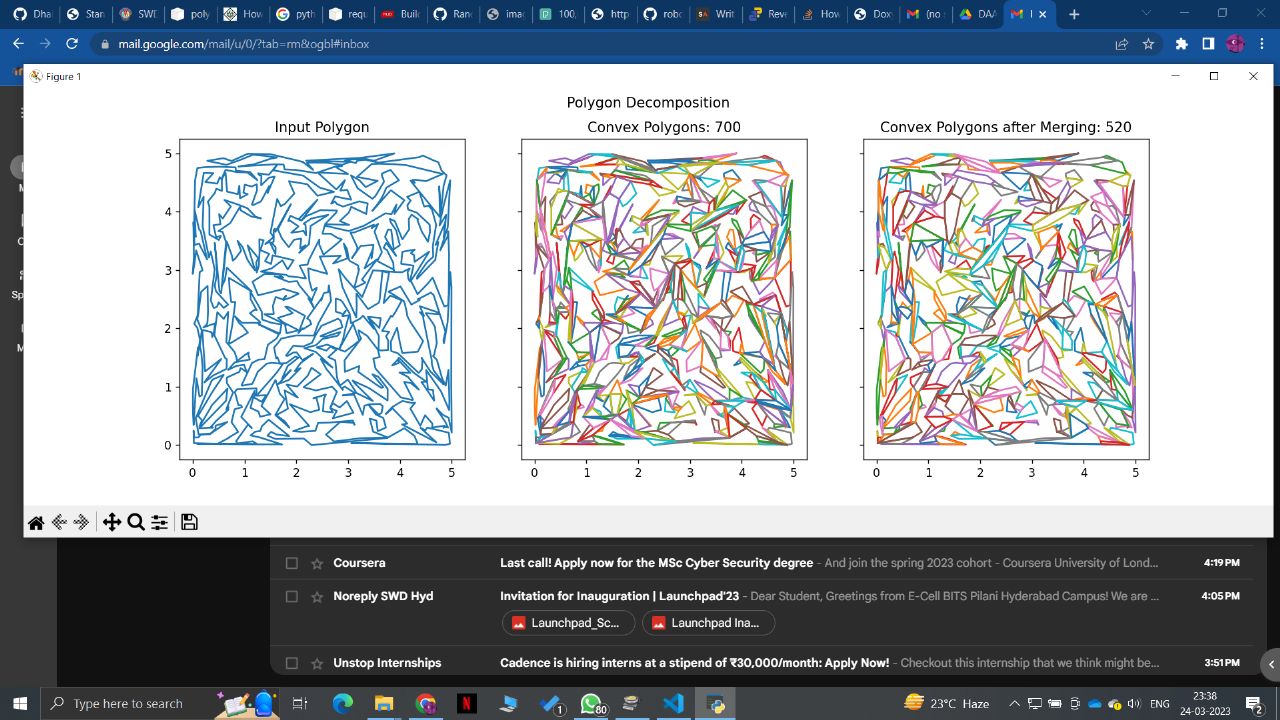
**TestCase 8:**

Number of Points = 200

**TestCase 9:**



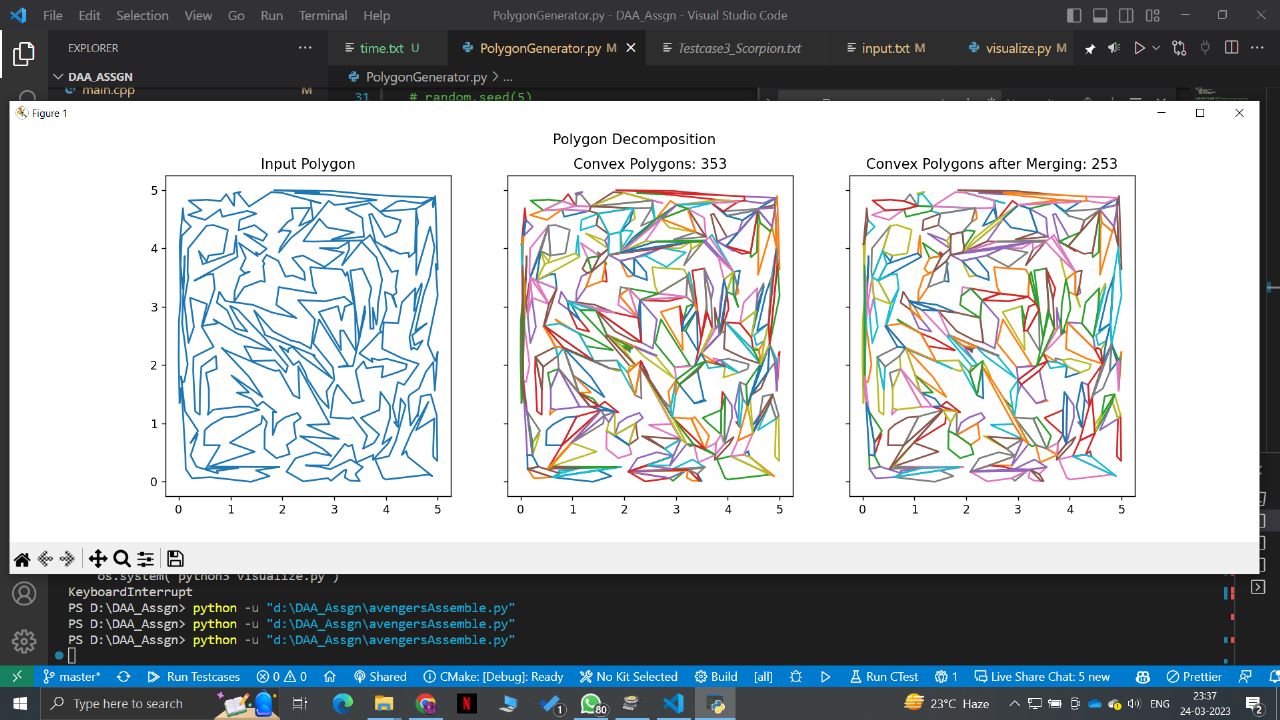
Number of Points = 300

**TestCase 10:**

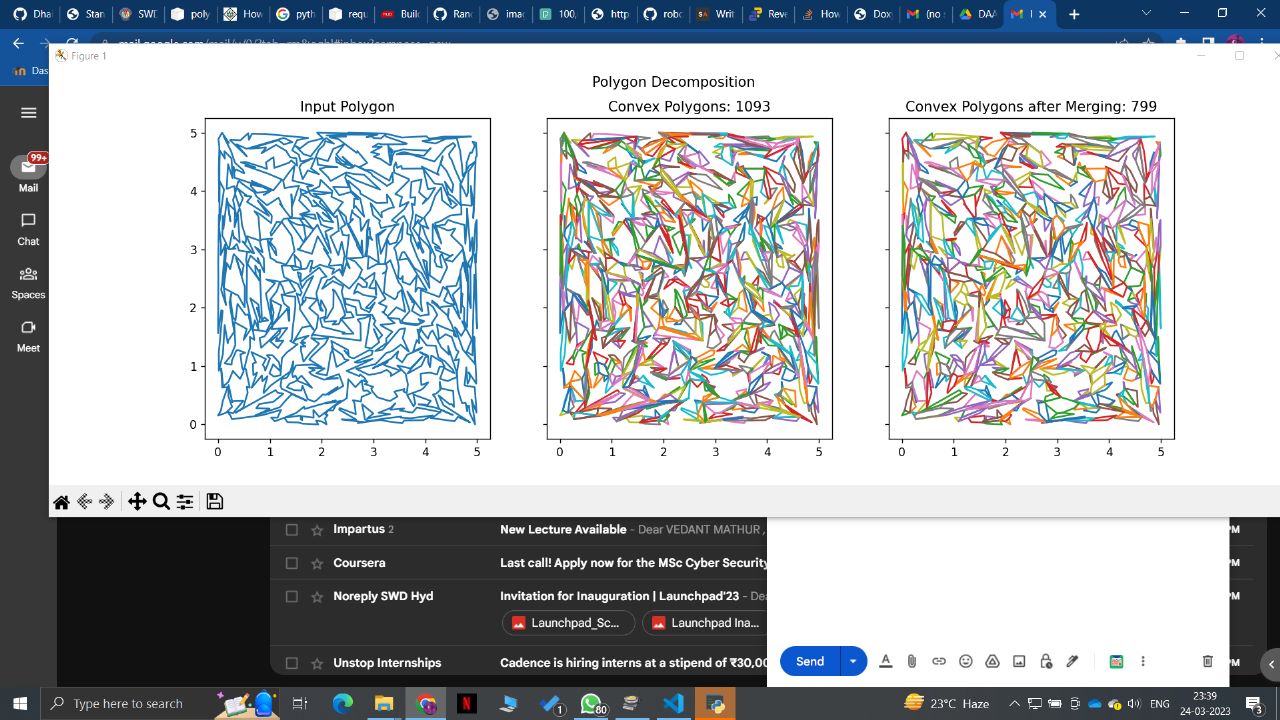
Number of Points = 500

**TestCase 11:**

Number of Points = 1000

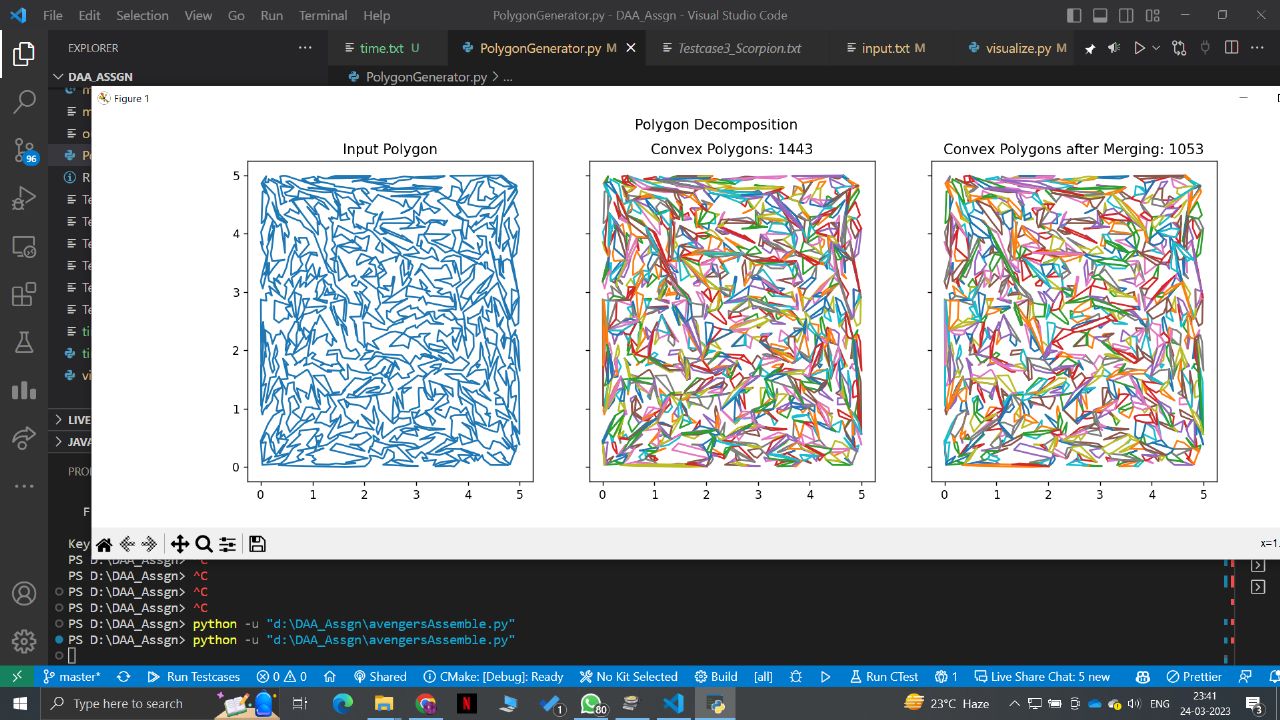


**TestCase 12:**

****

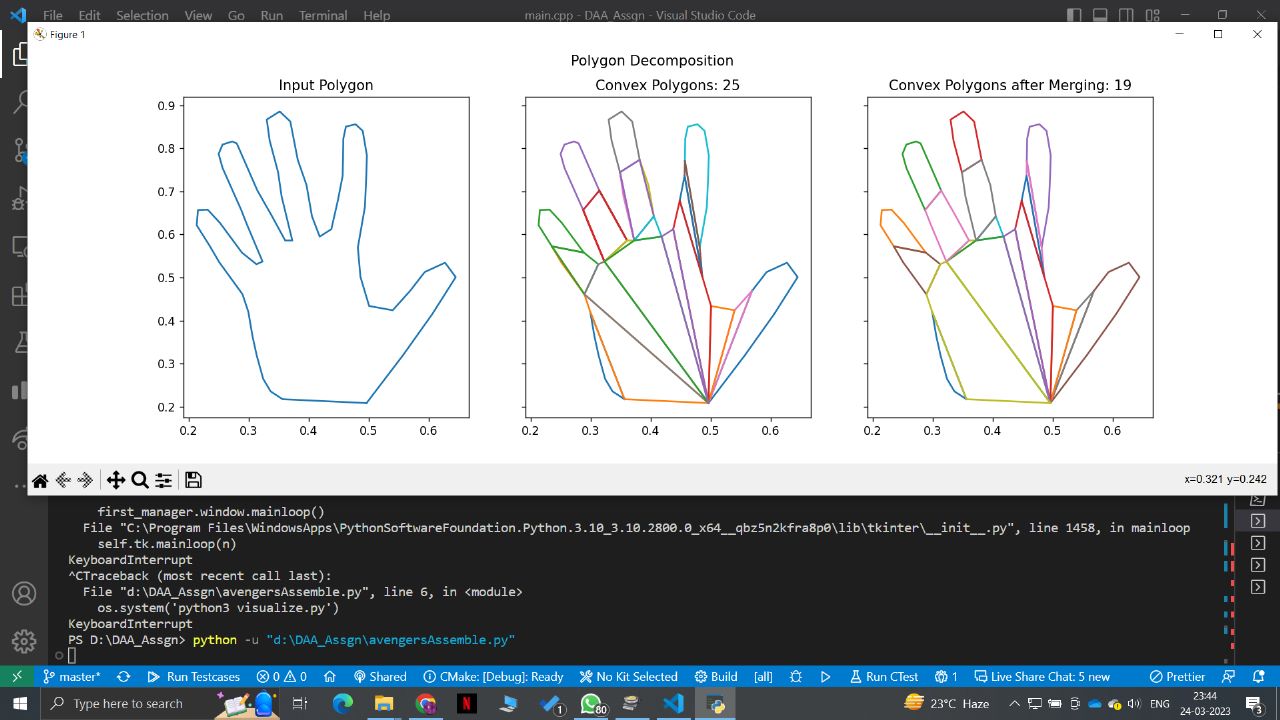
Number of Points = 1500

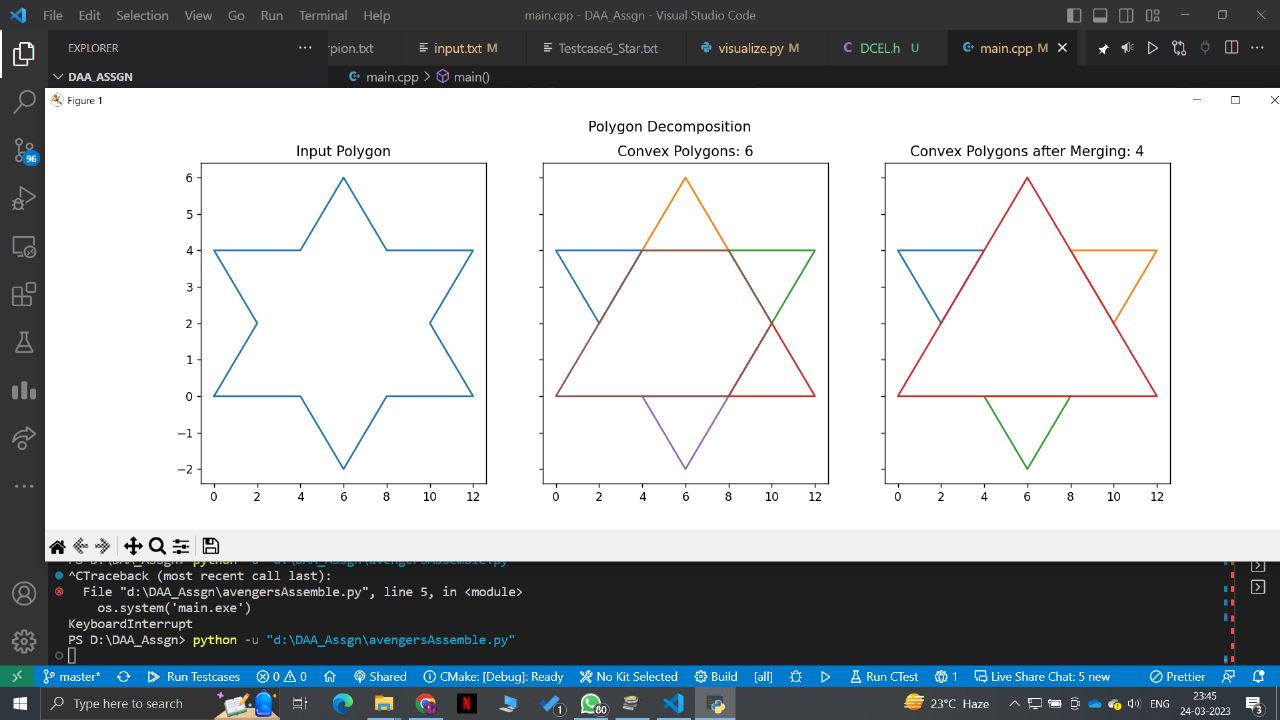
**TestCase 13:**

****

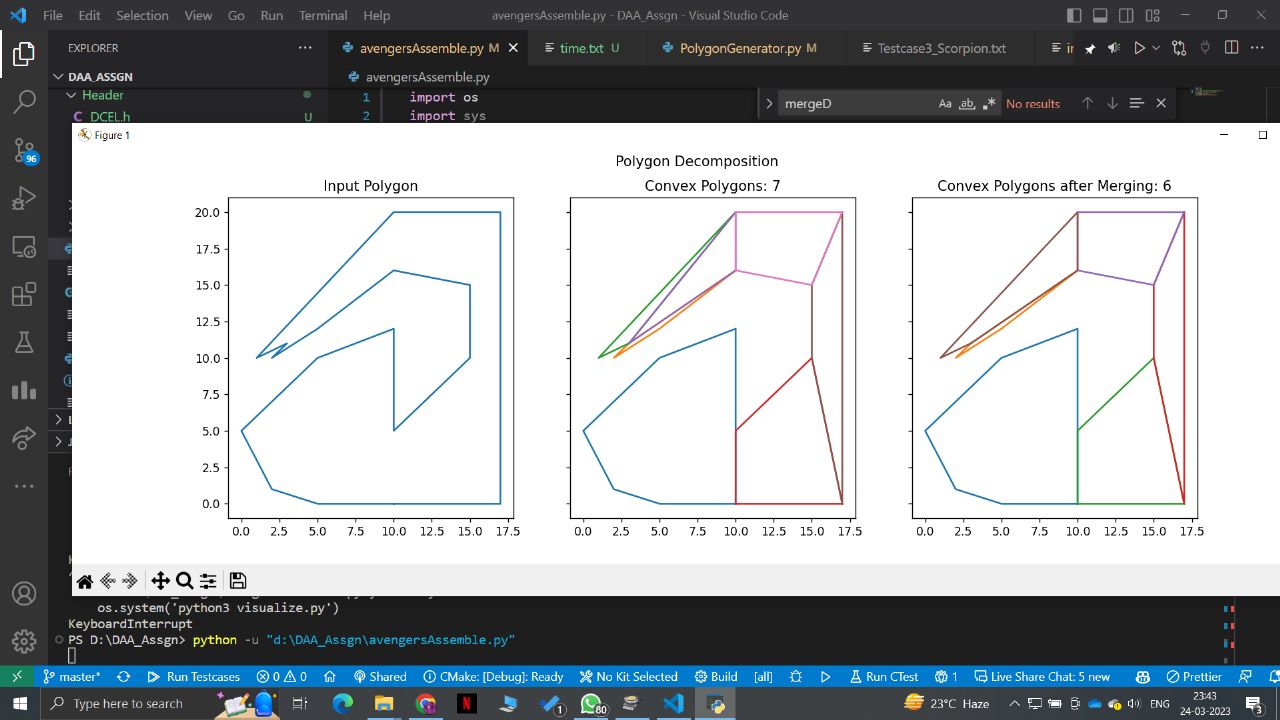
Number of Points = 2000

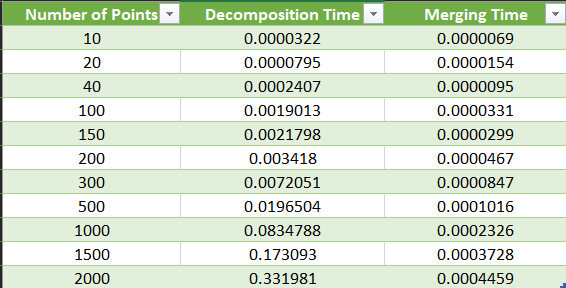
**TestCase Hand:**

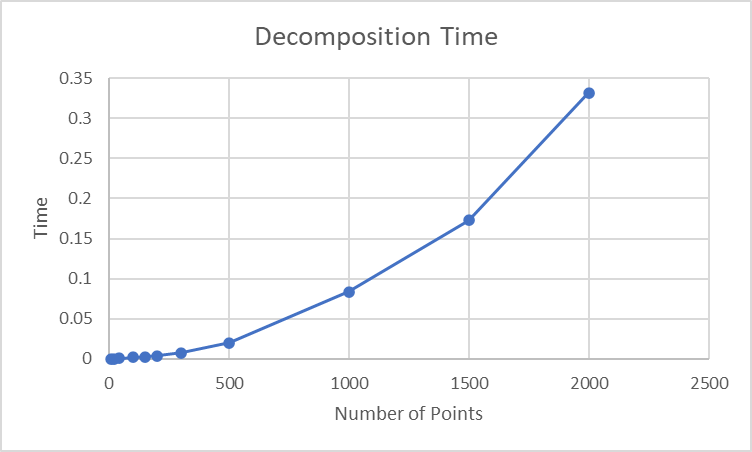
****

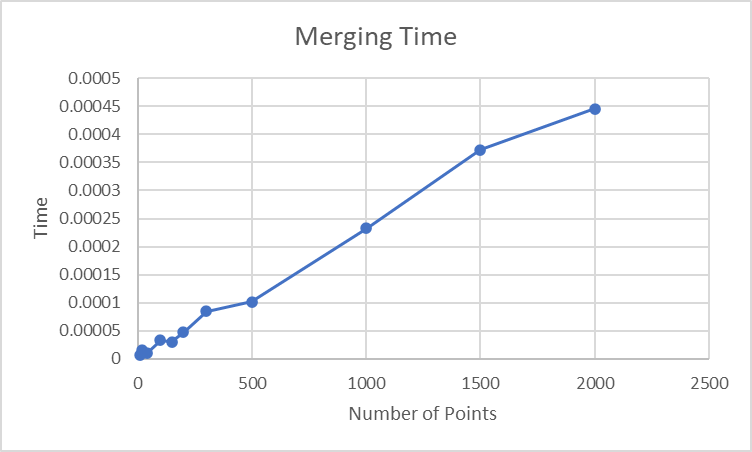
**TestCase Star:**

**TestCase Scorpion:**

****

****

****

****

The time complexity of the Decomposition Algorithm is O(n³). The outermost while loop runs until all the polygons are decomposed into a convex polygon, i.e., while n > 3, where n is the number of vertices of the original polygon without decomposition. Inside that, we add points to the list L, which, in the worst case I contains all the points. Since the time taken to calculate angles is constant, the inner while loop may take in the worst case O(n) time. The time taken to find the rectangle is O(n), and to find the list LPVS is also O(n). Finally, the time taken to check whether or not the point is inside the polygon is O(n) in the worst case, and finally to make the convex polygon, i.e. break the face into two convex faces is O(n) since we only need to create two new half-edges and change the values of all the half-edges.

The time complexity of the Merging Algorithm is O(n), where n is the number of diagonals in the convex polygon obtained by the decomposition algorithm. We already have the list of diagonals stored with us, so constant time is required to access those. Then we loop through all the diagonals and compare the angle between a1, a2, a3 and b1, b2, b3. Since all these points are being accessed in O(1) time using the HalfEdges and their twins, no additional time is incurred. Finally, if the diagonal is redundant and is to be removed, we use the constant time to delete the half-edge and connect the two convex polygons.

Therefore the total time required for merging is O(n), where n is the number of diagonals.

As evident from the above discussion and the time graph analysis, the merging takes more time than the decomposition algorithm. Also, it can be seen that with increasing values of n, the time taken by merging and decomposition increases.