

MOTH ERADICATION

BATCH-22

25-03-2022

Shri Vishnu Engineering College For Women,Bhimavaram

PROBLEM DESCRIPTION

- you must write a program that can take as input the locations of a traps in a region and output the locations of trap that lie on the perimeter of the region as well as the length of the perimeter.

SAMPLE INPUT

- 3
1 2
4 10
5 12.3
- 6
0 0
1 1
3.1 1.3
3 4.5
6 2.1
2 -3.2

SAMPLE INPUT CONTD

◇ 7
1 0.5
5 0
4 1.5
3 -0.2
2.5 -1.5
0 0
2 2
0

SAMPLE OUTPUT

- REGION 1 :

(1.0,2.0)-(4.0,10.0)-(5.0,12.3)-(1.0,2.0)

Perimeter length = 22.10

REGION 2:

(0.0,0.0)-(3.0,4.5)-(6.0,2.1)-(2.0,-3.2)-(0.0,0.0)

Perimeter length = 19.66

REGION 3:

(0.0,0.0)-(2.0,2.0)-(4.0,1.5)-(5.0,0.0)-(2.5,-1.5)-(0.0,0.0)

Perimeter length = 12.52

PROBLEM UNDERSTANDING

- ◇ we have to find a polygon that should have all the traps inside the polygon or on the outline of the polygon.
- ◇ To find a polygon of minimum perimeter and enclose all the traps we need to use **Convex Hull Algorithm**

APPROACH

- ◇ **Convex Hull:** The convex hull is the minimum closed area which can cover all given data points.
- ◇ **Graham Scan:** Graham's Scan algorithm will find the corner points of the convex hull. The time complexity is **$O(n \log n)$** .

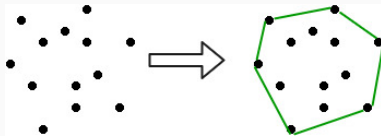


Figure 1: convex hull

PROJECT FLOW DIAGRAM

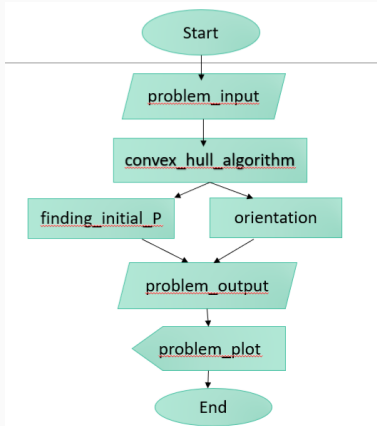


Figure 2: flow diagram

Day-wise Code Implementation:

- ❑ **21-03-2022** : Problem Understanding and Algorithm
- ❑ **22-03-2022** : Modular Approach for Solving problem Statement
- ❑ **23-03-2022**: Solving Modules of the Problem Description
- ❑ **24-03-2022** : Integrating all modules and Debugging
- ❑ **25-03-2022** : Final Code Testing and Review of Project

Project Gitlab Link:

<https://gitlab.com/20b01a0351/moth-eradiction.git>

Team Members Contribution:

533-

Worked on 2 modules and Testing

534-

Worked on 3 modules and Debugging

351-

Worked on Latex presentations and installation of libraries

350-

Worked on collecting resources and help for Latex Presentation

445-

Worked on 1 module and Gitlab commits

CHALLENGES

Challenges and How we Overcame:

1. Difficulty in Problem Understanding

To overcome this challenge , we referred Internet and understood the necessary algorithms to be used .

2. Modular Approach:

As it is very difficult to develop the code at once , we divided it into modules and integrated for better understanding.

3. Errors in code:

Understanding the problem in means of functionality and debugging it helped us to overcome the errors in the problem.

ALGORITHM TO FIND P0

- ◇ `int ymin = points[0].y`
- ◇ `min = 0`
- ◇ `for i in range(1,n)`
 - . `int y = points[i].y`
 - pick the bottom-most or choose the left
 - most the point in case of tie
 - . `if ((y < ymin) — (ymin == y points[i].x < points[min].x))`
 - . `ymin = points[i].y`
 - . `min = i`

TECHNICAL STACK:

- ◇ **PROGRAMMING LANGUAGE:** python 3
- ◇ **EDITOR:** Google Colaboratory
- ◇ **SITE:** Latex power point presentation
- ◇ **REPOSITORY:** Gitlab

***** THANK YOU *****