COL226 Assignment 1

Kuldeep Meena

February 23, 2021

Approach

1. Input Output Specification

The program takes input n which is followed by n coordinates (as X and Y coordinates respectively) such that the coordinates are sorted with respect to X coordinates(no condition on Y coordinates).

2. Some design decisions

a. Since it is common to take inputs of test cases with a line break I have opted for the same convention with a message for X and Y coordinates for each point.

b. I assume that we are concerned with the area bounded by points and the X axis

c. I assume that we are interested in area and not in integeration so both area above and below axis are taken positive although it is easy to expand the algorithm to integration as well by changing a small part.

d. The input points are bounded to be sorted with respect to X coordinate only so my design decision is to find area of that curve that is obtained by joined points that are sorted with Y if X is same which is more closer to a mathematical function than a figure where joining lines overalaps for a particular X

e. For the time being I have assumed that all my data is accomodable in 32 bits

Working of algorithm

The algorithm is such that it can be extended to any number of coordinates as it holds the invariant that after i points area is the area under the curve by the coordinates seen upto that point. The algorithm takes in care all possible cases i.e. points with y positive, y negative or a transition form positive to negative or zig zag curve or points that have 0 as X or Y. Algorithm works on standard mathematical notion of area calculation under line joined by points and proof of correctness is easy to see as most of the state updates are based

on mathematical observations for area under a curve.

The algorithm has constant space complexity i.e does not have an array of n numbers and hence can be extended to any n points curve. Also it does not have any sorting so time complexity is O(n).

INV: After i points $area is such that $area is the area bounded by (x1,y1).....(x1,yi) and the X axis

Some Test Cases

1. Test Case (All Positives):

n = 6
(-8,10),(-4,10),(-4,8),(-4,12),(6,9),(8,0)
Area = 150

2. Test Case (All Negatives):

n = 6
(-8,-10),(-4,-10),(-4,-8),(-4,-12),(6,-9),(8,0)
Area = 150

3 .Test Case (Positive to Negative):
n = 5
(-8,5),(-4,4),(-2,9),(1,-18),(3,0)
Area = 71.5

4. Test Case (Zig Zag):
n = 5
(-6,3),(-2,-9),(-1,-5),(3,15),(4,-5)
Area = 53.25

5. Same value at X and pos to neg (includes zero)
n= 7
(-5,2),(0,2),(0,3),(0,0),(0,5),(8,-9),(10,-5)
Area: 49.2857

6 .Same value at X and neg to pos (includes zero)
n = 7
(-5,-2),(0,-2),(0,-3),(0,0),(0,-5),(8,+9),(10,5)
Area: 49.2857

Testing Strategy
I have tried to do as exhaustive testing as possible and have also included some of the test cases that gives sense of the border cases for my algorithm. I have excluded the 32 bit crossing case as I have assumed that my data is within 32 bits . For my algorithm side cases include transition form negative to positive or vice versa and unsorted Y coordinates for a particular X, I have included both cases in test cases. Other than that I have also included test cases corresponding

to other trivial cases.