Assignment-1 Submission

Indian Institute of Technology Delhi

COL216: Computer Architecture

Name: Deepanshu **Entry Number: 2019CS50427**
Name: Prashant Mishra **Entry Number: 2019CS50506**

1 Introduction

The assignment is about writing a MIPS Assembly Program for obtaining the area under a curve formed by joining successive points by a straight line.

2 Approach taken to solve the problem

Following is the step wise description of the approach taken to solve the problem.

2.1 Writing pseudocode

We began by writing a pseudo code which is more inclined towards human understanding and high level programming.
We extensively covered all the possible scenarios in this pseduo code.

2.2 Breakdown of complex expressions

The next step involved breakdown of complex mathematical expressions into multiple line of simpler commands.
For example: a = (a + b) + (c + d) will be broken into three statement as

1. a = a + b
2. c = c + d
3. a = a + c

This will be done keeping in mind that we have to eventually code in MIPS.
Also, some of the observatory modifications were done like we don't need to store all the coordinates in an array. Rather, we can compute the area simultaneously while taking the input. Thereby reducing the space complexity from O(n) to O(1) without disturbing the time complexity.

2.3 Programming in MIPS

After modifying the pseudo code as stated above, we started the actual coding part where we converted statements from above simplified pseudo code into MIPS commands of loading, writing, adding etc.

3 Testing Strategy

The testing strategy had 3 layers.

3.1 Testing the pseudo code

In this, we took the pseudo code and programmed it in a high level language (C++). We then tested it against various input and confirmed its correctness. We also stored the values corresponding to some test cases.

3.2 Testing the MIPS code

Here, we took the same test cases that we used in part 1 and matched the output with the output from part 1. This ensured that the code is converted correctly into MIPS language.

3.3 Choosing the test cases

We tested against following possibilities:

1. Whole graph lies above x-axis.
   Eg: n = 3 : (1,1) , (2,2) , (3,3). Area: 4.00
   Eg: n = 4 : (2,3) , (4,5) , (7,8) , (8,4). Area: 33.50

2. Whole graph lies below x-axis.
   Eg: n = 3 : (1,-1) , (2,-2) , (3,-3). Area: 4.00
   Eg: n = 3 : (-3,-5) , (3,0) , (4,-3). Area: 16.50

3. Graph went from above x-axis to below x-axis and stayed below x-axis.
   Eg: n = 3 : (1,1) , (2,-2) , (3,-3). Area: 3.33
   Eg: n = 2 : 2,3) , (4,-3). Area: 3.00

4. Graph went from below x-axis to above x-axis and stayed above x-axis.
   Eg: n = 3 : (1,-1) , (2,2) , (3,3). Area: 3.33
   Eg: n = 4 : (-3,-3) , (0,0) , (3,0) , (5,7). Area: 11.50

5. Graph first went from above x-axis to below x-axis and then went again above x-axis.
   Eg: n = 4 : (1,1) , (2,-2) , (3,-3) , (4,4). Area: 5.119
   Eg: n = 3 : (0,4) , (6.-2) , (12,4). Area: 20.00

6. Graph first went from below x-axis to above x-axis and then went again below x-axis.
   Eg: n = 4 : (1,-1) , (2,2) , (3,3) , (4,-4). Area: 5.119
   Eg: n = 3 : (-1,-1) , (1,1) , (3,-1). Area: 2.00

7. Some combination of the above 5 cases.
   Eg: n = 6 : (1,1) , (2,2) , (4,-2) , (5,-1) , (7,2) , (8,1). Area: 8.166
   Eg: n = 5 : (1,1) , (3,4) , (5,3) , (6,7) , (9,5). Area: 35.00
   Eg: n = 7 : (0,0) , (2,3) , (5,0) , (7,-2) , (9,0) , (9,3) , (10,0). Area:13.00
   Eg: n = 10 : (-3,-4) , (-1,0) , (2,0) , (2,1) , (2,4) , (3,4) , (6,4) , (10,8) , (18,0) , (20,0). Area: 76.00

8. Some special case.
   (a) Single coordinate as input.
       Eg: n = 1 : (1,1)
       The code gives 0 as the output.
   (b) $n \leq 0$
       The program throws an error.

9. Some portion of graph has two consecutive points with same x coordinate i.e it forms a vertical axis.
   In that case, the code considers that area to be 0.
   Eg: n = 2 : (2,-3) , (2,6). Area: 0.00
   Eg: n = 4 : (-3,-4) , (2,0) , (2,5) , (4,5). Area: 20.00