Rectangles Project D2S
Long Term Internship Task

Mohamed Okasha Mohamed

Computer and Systems Department, Faculty of Engineering

Ain Shams University

31/10/2019

Abstract

This report describes all the work related to the task. The used algorithm, classes and platform details are described in addition to all the testing procedure and results with all information related to the memory and execution time.

Table of Contents

Contents

Abstract 2

Table of Contents 3

List of Figures 4

List of Tables 5

1 Introduction 6

2 Program information 6

2.1 Used Platform 6

2.2 Algorithm Details 6

2.2.1 Main Function Flowchart 7

2.2.2 Algorithm Flowchart 8

2.2.3 Pseudocode 9

3 Testing Methodology 9

4 Results and discussion 11

5 Screenshots of The Output 13

List of Figures

1. Flowchart to describe the procedure of the main function 7

2. Flowchart to describe the procedure of the grouping algorithm 8

3. Dataset 3 drawn by AutoCAD 9

4. Program output when testing the touching rectangles case 10

5 program output when testing the identical rectangles case 10

6. program output for the additional testing dataset 10

7. Error checking output 11

8. Visual Studio output vs eclipse output for datasets 13,14,15 13

9. eclipse output for datasets 9,10,11,12 13

10. profiler output for datasets 13,15 14

List of Tables

1. results table 12

1 Introduction

In this task, I had the idea to use a divide and conquer method to obtain the biggest group first then recall the function recursively to find the biggest group on the remaining rectangles and so on. but I think this will give the same results as the brute force method “which I already used and was too much faster”. The brute force method is checking each rectangle with the next all rectangles in a vector. The algorithm will be discussed briefly in the methodology section.

2 Program information

In this section, I will describe briefly all the information related to the program

2.1 Used Platform

This is all the details regarding the used platform:

1. Operating System: Windows 10
2. Processor: intel core i7 – 7500U
3. IDE: visual studio for memory calculation and profiling – VS code & eclipse as Editors

2.2 Algorithm Details

In this section I will describe the used algorithm by flowcharts and pseudocode. Note that some of the details of the flowchart might be not clear so I will attach pdf files with the flow charts

2.2.1 Main Function Flowchart

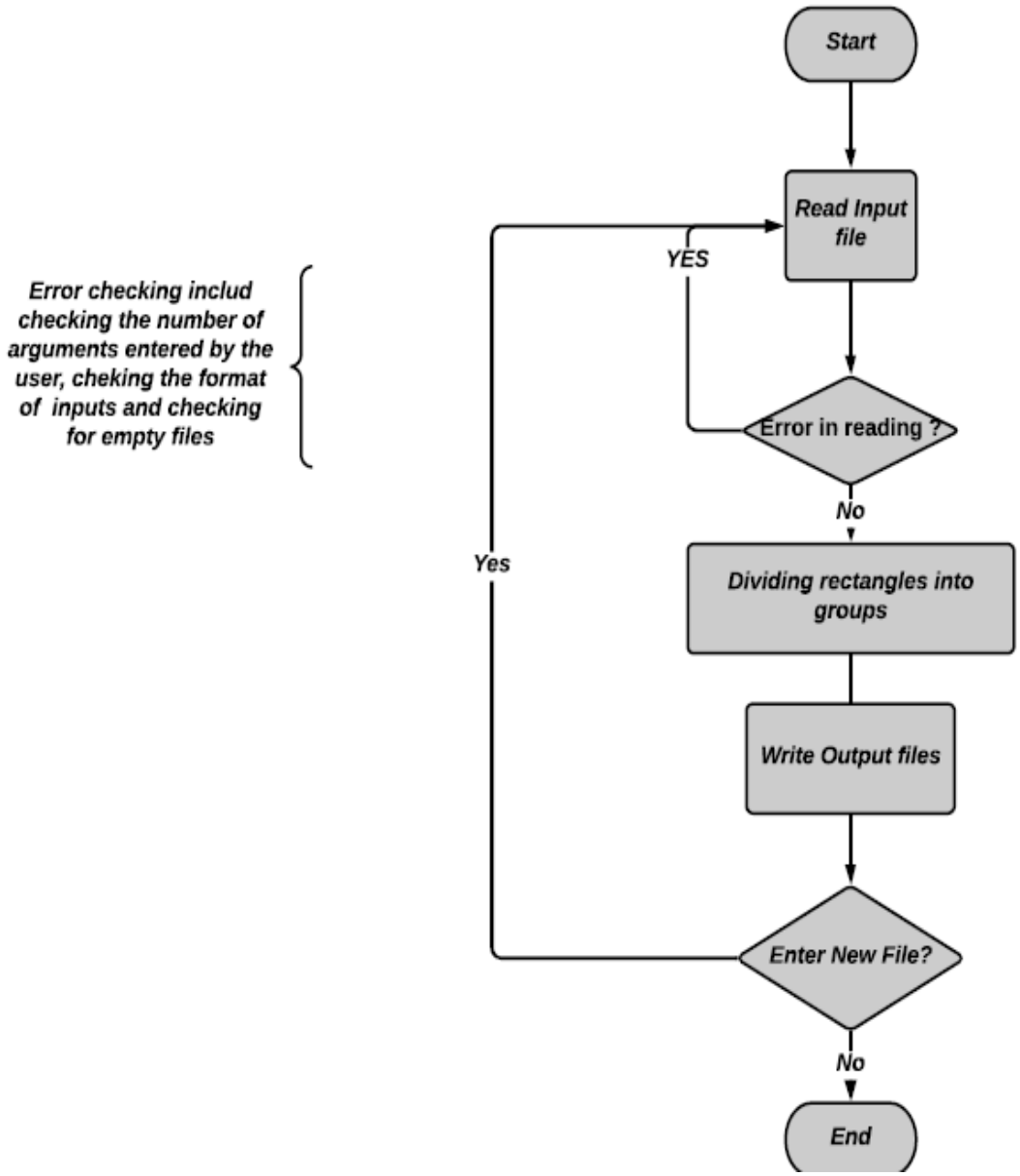


Figure 1: flowchart to describe the procedure of the main function

2.2.2 Algorithm Flowchart

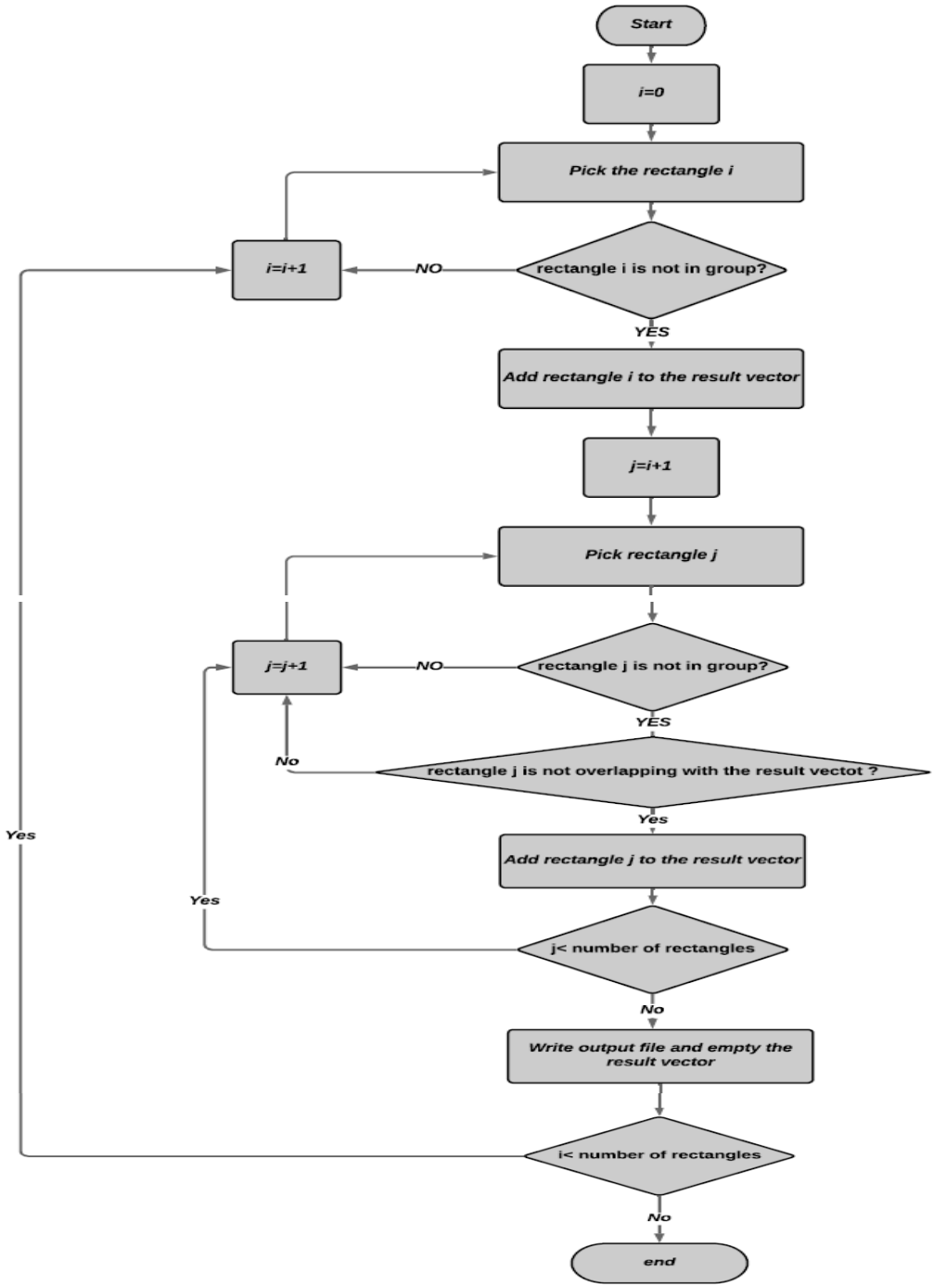


Figure 2: flowchart to describe the procedure of the grouping algorithm

2.2.3 Pseudocode

The following code describes the algorithm used to divide the rectangles into groups.

```
for i = 0 : Size of rectangles vector
    if rectangle of i is not in group
        add rectangle i to the result vector
        mark rectangle i as taken

    for k = i+1 : size of rectangles vector
        if rectangle k isn't overlapping with all rectangles in the result vector
            add rectangle k to the result vector
            mark rectangle k as taken

    if the result vector started with rectangle i is not empty
        output this vector to an output file
        empty the result vector so as to be used again
```

3 Testing Methodology

For testing purposes, I tried some of the basic cases of the rectangles overlapping in addition to testing complete data sets. The testing methods include:

1. Drew the first 3 datasets and testing them manually

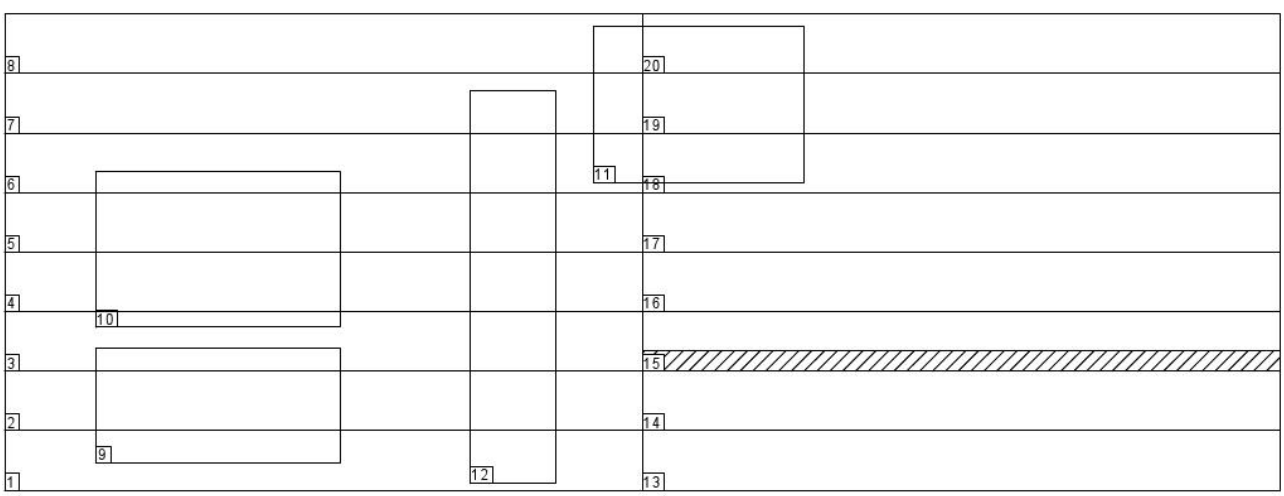


Figure 3: Dataset 3 drawn by AutoCAD

2. Rectangle above rectangle

3. Rectangle beside rectangle

4. Touching rectangles

```
{
    rectangle temp1(1, 1, 3, 3);
    rectangle temp2(3, 3, 5, 5);
    if (temp1.dontOverlap(temp2))
        cout << "not overlapping";
    else
        cout << "overlapping";
    /* ... */
```

```
not overlapping
G:\Engineering\D2S-Project\Debug\D2S-Project.exe (process 10060) exited with code 0
To automatically close the console when debugging stops, enable Tools->Options->Deb
ally close the console when debugging stops.
Press any key to close this window . . .
```

```
int main()
{
    rectangle temp1(1, 1, 3, 3);
    rectangle temp2(1, 3, 3 , 5);
    if (temp1.dontOverlap(temp2))
        cout << "not overlapping";
    else
        cout << "overlapping";
    /*      */
```

Microsoft Visual Studio Debug Console

```
not overlapping
G:\Engineering\D2S-Project\Debug\D2S-Project.exe (process 15576) exited with code 0.
Press any key to close this window . . .
```

Figure 4: program output when testing the touching rectangles case

5. Two rectangles with the same corner points

```
rectangle temp1(1, 1, 3, 3);
rectangle temp2(1, 1, 3 , 3);
if (temp1.dontOverlap(temp2))
    cout << "not overlapping";
else
    cout << "overlapping";
```

Microsoft Visual Studio Debug Console

```
overlapping
G:\Engineering\D2S-Project\Debug\D2S-Project.exe (process 16972) exited with code 0.
Press any key to close this window . . .
```

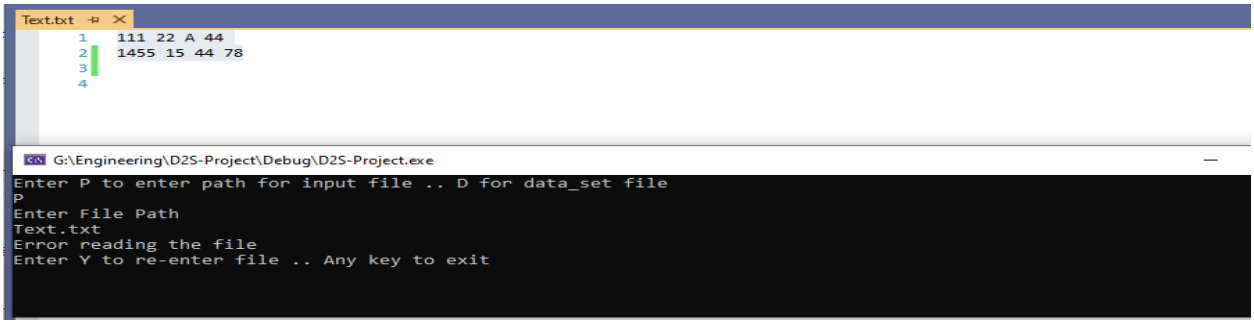
Figure 5: program output when testing the identical rectangles case

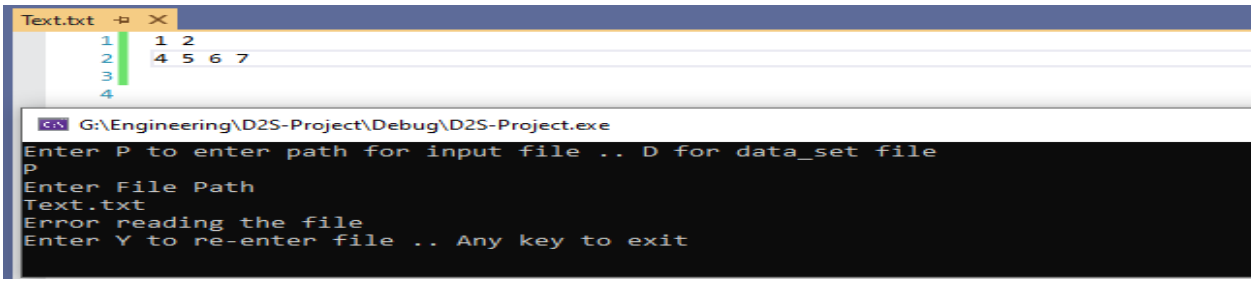
6. Tried a new dataset that contains the sum of all the first 15 dataset's rectangles to determine the time and memory usage for the 40k rectangles

```
Enter P to enter path for input file .. D for data_set file
D
Enter data_set number
17
Number Of Inputs = 40750
Number Of Groups = 17
Time Consumed = 10.8573 sec
```

Figure 6 : program output for the additional testing dataset

7. Tried the following test datasets for error checking (few inputs or wrong variables format)





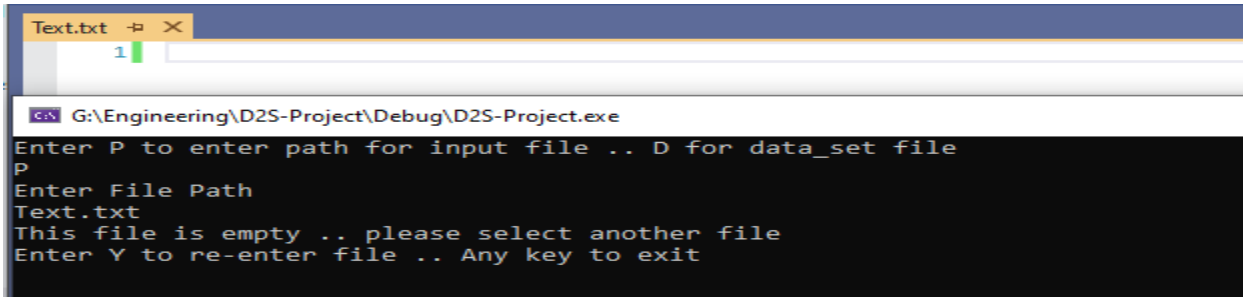


Figure 7: Error checking output

<u>Notes:</u>

1. I considered the touching rectangles in a single point or in a line are not overlapping.
2. Dataset 16 was read successfully in 11 seconds, but the processing part took much time without any results

4 Results and discussion

The following tables provides all the results of testing the algorithm on your datasets. The algorithm passed all the datasets except for the huge one. As you will see in the screenshots, some of

the runtimes determined from eclipse IDE are different from the runtimes determined from visual studio IDE. Also, visual studio adds some additional instructions for profiling purposes which increase the runtime of the algorithm.

Data set	Number of inputs	Number of output groups	Runtime (sec)	Memory usage
data_set_1	5	1	0.0009	828 KB
data_set_2	7	2	0.0019	852 KB
data_set_3	20	2	0.0019	852 KB
data_set_4	39	3	0.0049	855 KB
data_set_5	77	4	0.0049	855 KB
data_set_6	136	5	0.0069	848 KB
data_set_7	216	5	0.0069	860 KB
data_set_8	460	6	0.0119	860 KB
data_set_9	741	12	0.1114	896 KB
data_set_10	981	7	0.0984	896 KB
data_set_11	5793	7	0.2999	1.2 MB
data_set_12	6775	7	0.3793	1.2 MB
data_set_13	7538	7	0.4573	1.2 MB
data_set_14	8774	8	0.6081	1.4 MB
data_set_15	9188	7	0.6660	1.5 MB
data_set_16				

Table 1: The results table

5 Screenshots of The Output

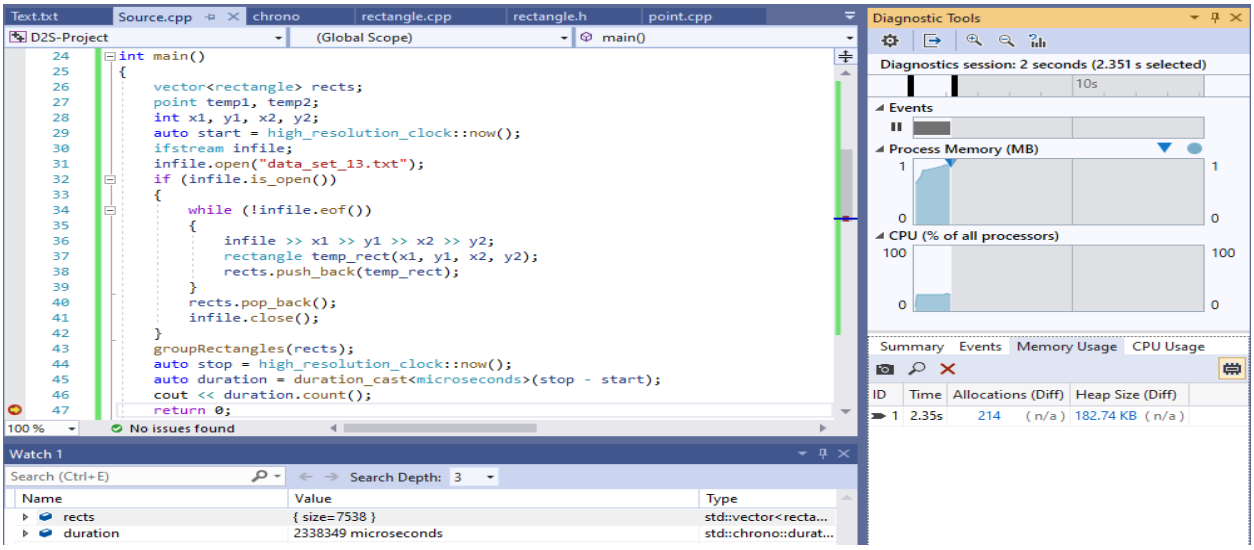
```
Enter Y to re-enter file .. Any key to exit
Y
Enter P to enter path for input file .. D for dat
D
Enter data_set number
13
Number Of Inputs = 7538
Number Of Groups = 7
Time Consumed = 1.15023 sec
Enter Y to re-enter file .. Any key to exit
Y
Enter P to enter path for input file .. D for dat
D
Enter data_set number
14
Number Of Inputs = 8774
Number Of Groups = 8
Time Consumed = 1.53312 sec
Enter Y to re-enter file .. Any key to exit
Y
Enter P to enter path for input file .. D for dat
D
Enter data_set number
15
Number Of Inputs = 9188
Number Of Groups = 7
Time Consumed = 1.51537 sec
```

```
Enter Y to re-enter file .. Any key to exit
Y
Enter P to enter path for input file .. D for data_se
D
Enter data_set number
13
Number Of Inputs = 7538
Number Of Groups = 7
Time Consumed = 0.457398 sec
Enter Y to re-enter file .. Any key to exit
Y
Enter P to enter path for input file .. D for data_se
D
Enter data_set number
14
Number Of Inputs = 8774
Number Of Groups = 8
Time Consumed = 0.608163 sec
Enter Y to re-enter file .. Any key to exit
Y
Enter P to enter path for input file .. D for data_se
D
Enter data_set number
15
Number Of Inputs = 9188
Number Of Groups = 7
Time Consumed = 0.666077 sec
```

Figure 8: Visual Studio output vs eclipse output for datasets 13,14,15

```
Enter Y to re-enter file .. Any key to exit
Y
Enter P to enter path for input file .. D for data_set file
D
Enter data_set number
9
Number Of Inputs = 741
Number Of Groups = 12
Time Consumed = 0.111462 sec
Enter Y to re-enter file .. Any key to exit
Y
Enter P to enter path for input file .. D for data_set file
D
Enter data_set number
10
Number Of Inputs = 981
Number Of Groups = 7
Time Consumed = 0.098422 sec
Enter Y to re-enter file .. Any key to exit
Y
Enter P to enter path for input file .. D for data_set file
D
Enter data_set number
11
Number Of Inputs = 5793
Number Of Groups = 7
Time Consumed = 0.299923 sec
Enter Y to re-enter file .. Any key to exit
Y
Enter P to enter path for input file .. D for data_set file
D
Enter data_set number
12
Number Of Inputs = 6775
Number Of Groups = 7
Time Consumed = 0.379363 sec
```

Figure 9: eclipse output for datasets 9,10,11,12



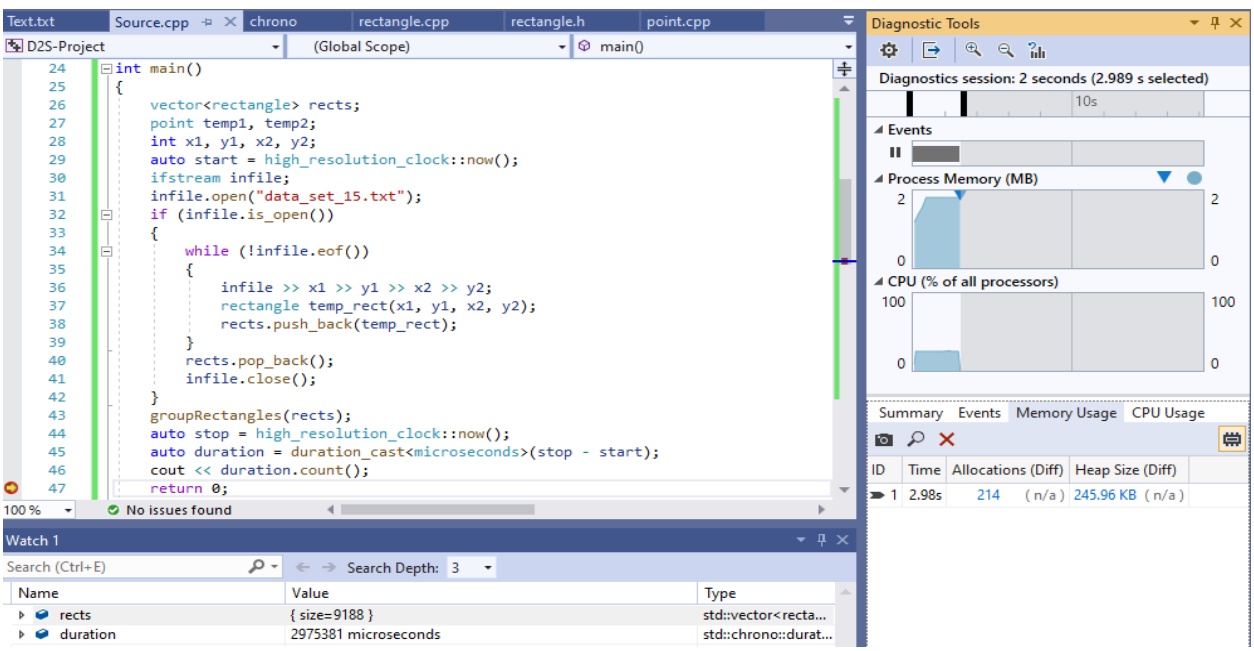


Figure 10: profiler output for datasets 13,15