Harshit Verma
ID: 200978548

QuadTree Data Report

Contents

Harshit Verma
ID: 200978548

The modular structure of the code

main.c: This includes the main functions, which all execute from the program during the makefile and allow the program to be executed. I have included the libraries stdio.h, string.h, stdlib.h, math.h, stdbool.h and other header files i.e node_structure.h and node_function.h

- int main: Inside int main I have included it to point to an empty head inside the node structure and then went to include the **makeChildren** function which will split the node inside the head into 4 children, level 0,1,2 and 3. I have also included all the functions which I will use throughout the program in other modules, such as **initialiseTheList, LinkedList, writeLinkedTree**, I have also included **writeTree** and the destroy node below that. (That is the part of the exercise given in the second week)

destroy.c: Inside here I have also included the libraries as same as the other modules which include: stdio.h, string.h, stdlib.h, math.h, stdbool.h and other header files i.e node_structure.h and node_function.h
This method will allow the quadtree to be destroyed after use.

- void destroyNode: This is the function which will help destroy the quadtree after its use inside the gnuplot. I have included a if statement to point to every node, and to check if the node is empty then just free that node and if it is not empty then simply destroy the node finishing this function.

Linked_LeafList.c: This is the linkedTree.c, Inside here I have also included the libraries as same as the other modules which include: stdio.h, string.h, stdlib.h, math.h, stdbool.h and other header files i.e node_structure.h and node_function.h.
Iniside this module I used :-

- initialiseTheList – initialise the new list for the leafs

Harshit Verma
ID: 200978548

- addNode – using this function to add node
- LinkedList – create the linked list for the leaf nodes
- writeLinkedTree – this function opening the file and witing inside the file.

- WriteLinkedNode – writing all nodes inside the file

- growTree - Using the leaf-node list add one level to every leaf node in the tree, growing it uniformly overall by one level.

- dataFunction - this function returns a value at a physical location in the quadtree. The choice value is 0,1 or 2 and selects different underlying data models.You should choose one value for testing your code. Once you have validated your code try the other values

- indicator - this function returns **false** we should add children to the current node. If it returns **true** we do not add children.
  We should pass through the current set of leaf nodes in the tree and use the **indicator()** function above to decide whether to add children to that node. For every complete pass through the leaf nodes we should record the number of **false** results.

- generateQuadtree - Generating a data-dependent Quadtree

make_node.c: Inside here I have also included the libraries as same as the other modules which include: stdio.h, string.h, stdlib.h, math.h, stdbool.h and other header files i.e node_structure.h and node_function.h.

- makeNode: This will allow the node to increment and go through every level to check to see if it has children and then also create a new level whilst it goes throught the list.
- makeChildren: This will allow it to split the node into children and then create them for each and every level that requires them.

output.c: This includes the header and libraries from before inside the file. This will in summary print out and show the tree along with writing out to the file, as well as the functions. (This is the part of the exercise given in week 2)

- void writeTree: Open and write to the out gnu file
- void writeNode: printout function to print out the nodes from each level

printout.c: This includes all of the Printout functions inside including printing out every level of the tree structure, along with implementing every level inside the out file and printing to the screen, inside the gnuplot program.

- printOut: This will simply print out the whole functions and level of nodes onto the out file, inside the gnu.

node_structure header: This allows the modifcation of the node as well as its creation link between other modules, this will allow the header file to be accessed in other module files aswell.

- node data structure: This is the data node structure, allows for values to be inputed and be used.
- type def struct qnode Node: This is the struct for the other tree values.
- struct linked_list: This is the linked_list for the whole of the program which will use it for the linking.

node_function header: This allows connection between other functions as well as setting certain values with the functions and variables, as well as the head file which will be included in other module files. This includes the call of all the functions listed below which will be used inside the other modules and then be called inside all of the other modules.

- void inititaliseTheList
- void LinkedList

Harshit Verma
ID: 200978548

- void writeLinkedTree
- void writeLinkedNode
- void printOut
- void makeChildren
- void writeTree
- void writeNode
- void destroyNode
- Node *makeNode

A suitable test plan for the implementation tasks described above

TRIVIAL TESTS

The code aim is to create a quad tree, I have envisioned this as it to run perfectly, with few errors which I can fix, this is the case, it worked! And my program has been completed to full specification

- Create the list, write the Linked tree and create a function which open the file and write in it:
  First I will give a function which will create the list of leafs nodes and linked all nodes to it. And then inside the if statement I will call the another function which will add the node to the list and otherwise it will walk through to leafs.
  When I ran the program it worked perfectly.
  I used these functions to run the program properly:-
  Node *makeNode(): This function make the node

  void makeChildren(): This function make the children.

  void printOut(): It will print out the quadtree on the gnuplot

  void destroyNode(): It will destroy the node that is free the node void writeLinkedTree(): This function writing the linked list

void addNode(): It is adding the function to the linked list

void initialiseTheList(): Initialising the list for the leafs

void writeLinkedNode(): This function is opening the file and creating the tree in that file.

void LinkedList(): This function create the linked the list

- <u>Growing QuadTree by 1 Level:</u>
  What I want when my Quadtree grows by 1 level, it should work and also increment say a level 2 hence creating 7 nodes.
  I will create the a function which will increase the level and print in the gnuplot.
  After testing this it worked perfectly.

  void growTree(): This function increase the level of the quadtree

- <u>Data Dependent QuadTree:</u>
  In valueTree.c, the function is already created 'dataFunction' which returns a value at a physical location in the quadtree, there are choices given like 0,1,2 which will change the locations in thee quadtree. And another function 'indicator' provided in the valueTree.c which will add the children to current node if it returns false and if it returns true then it will not add the children.
  I will create the another function inside the if statement which will basically call the 'indicator' function and equal it to false. And then call the function which will make the children.

  Void dataFunction(): his function returns a value at a physical location in the quadtree. The choice value is 0,1 or 2 and selects different underlying data models.You should choose one value for testing your code. Once you have validated your code try the other values

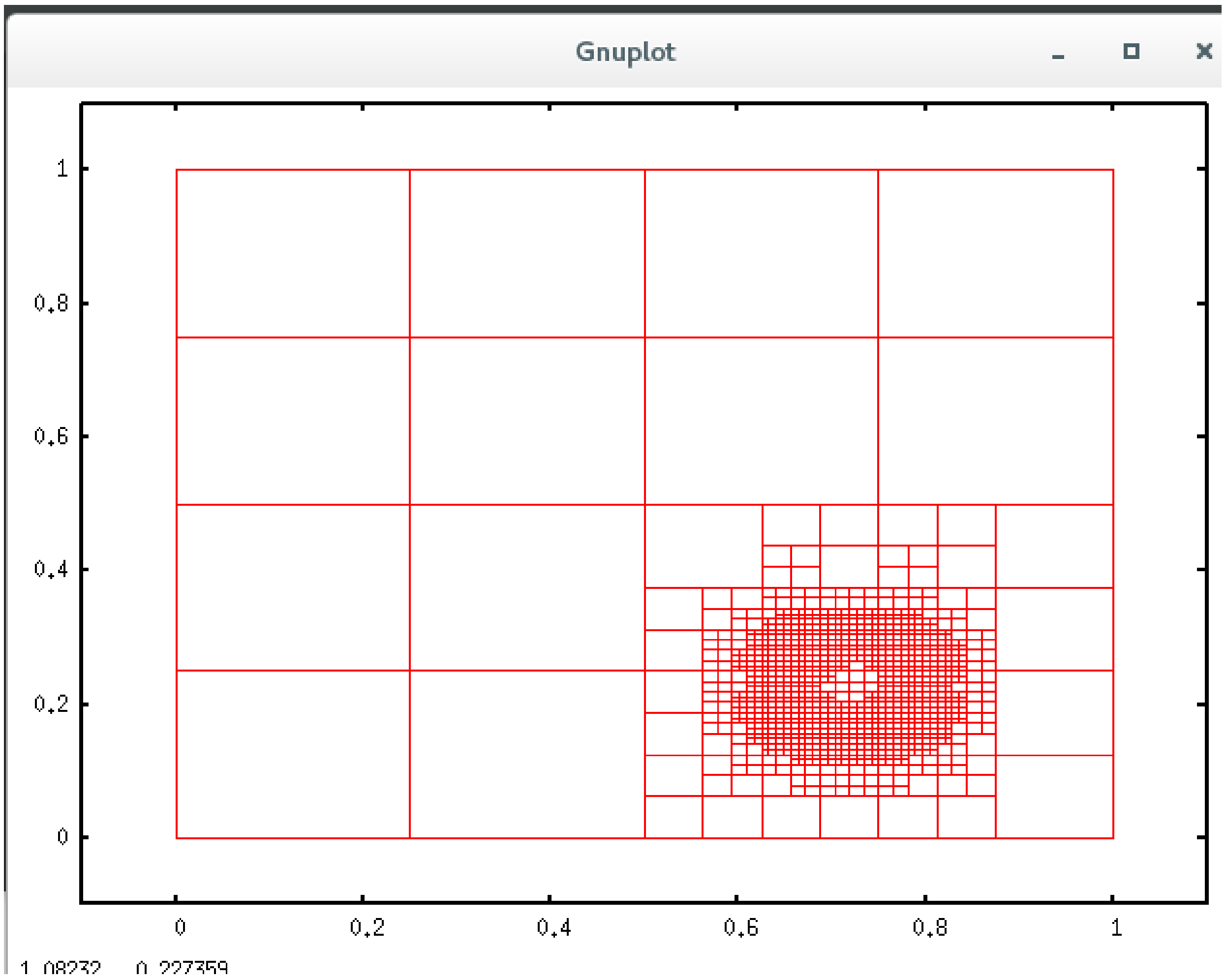
Harshit Verma
ID: 200978548

bool indicator() - this function returns **false** we should add children to the current node. If it returns **true** we do not add children.
We should pass through the current set of leaf nodes in the tree and use the **indicator()** function above to decide whether to add children to that node. For every complete pass through the leaf nodes we should record the number of **false** results.

void generateQuadtree(): This function is generating the data dependent quadtre

NON TRIVIAL TESTS

- Test Bounderies of applications:
  I have tested my code and it is working properly. When I changed the tolerance to 0.05 and choice to 2 it was showing



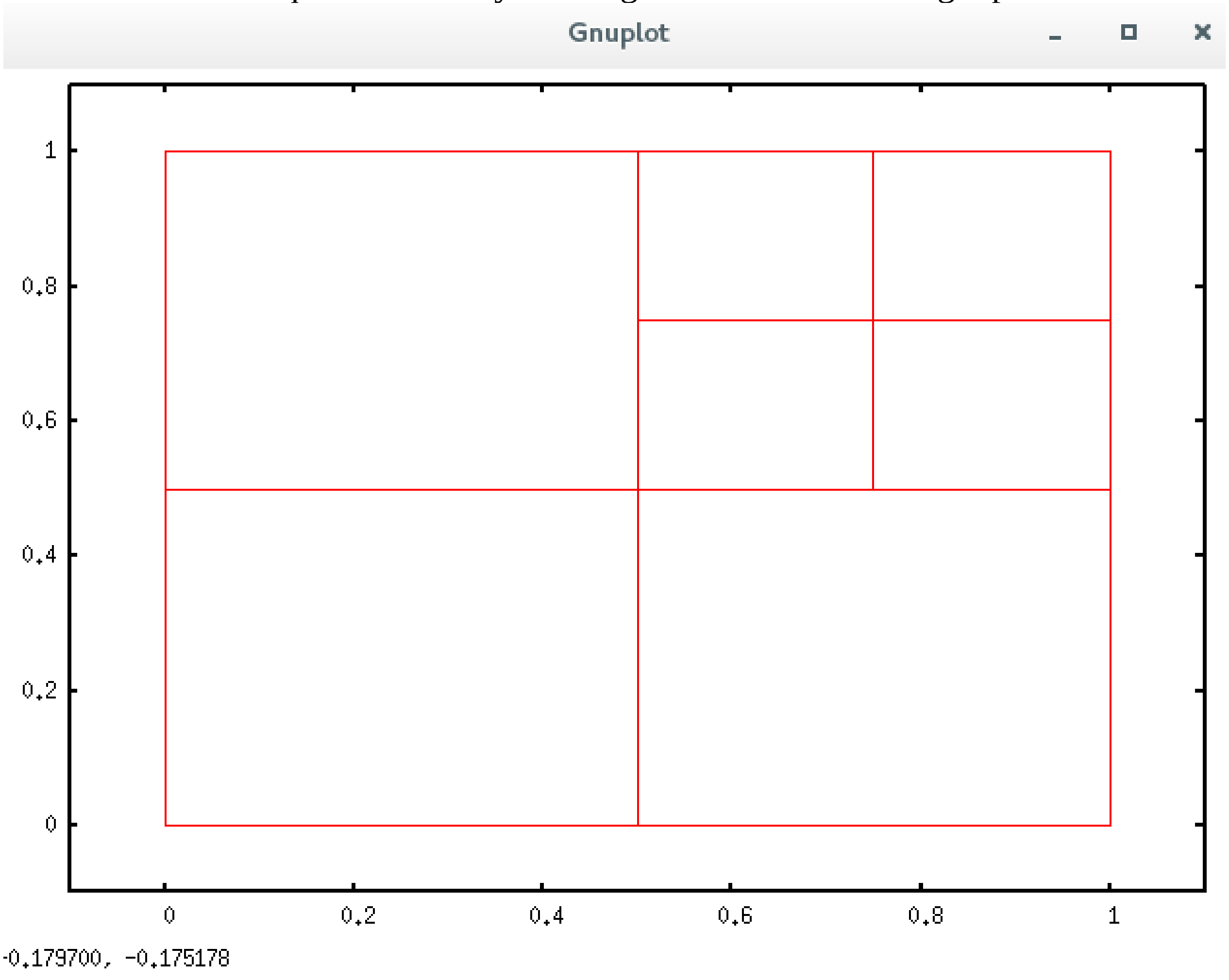
Harshit Verma
ID: 200978548

- When I tested my code with tolerance 0.005 and choice to 0 , it took so long time to print on the gnuplot and sometimes terminal had crashed.

<u>**Results from the non-trivial tests that show that your application works**</u>

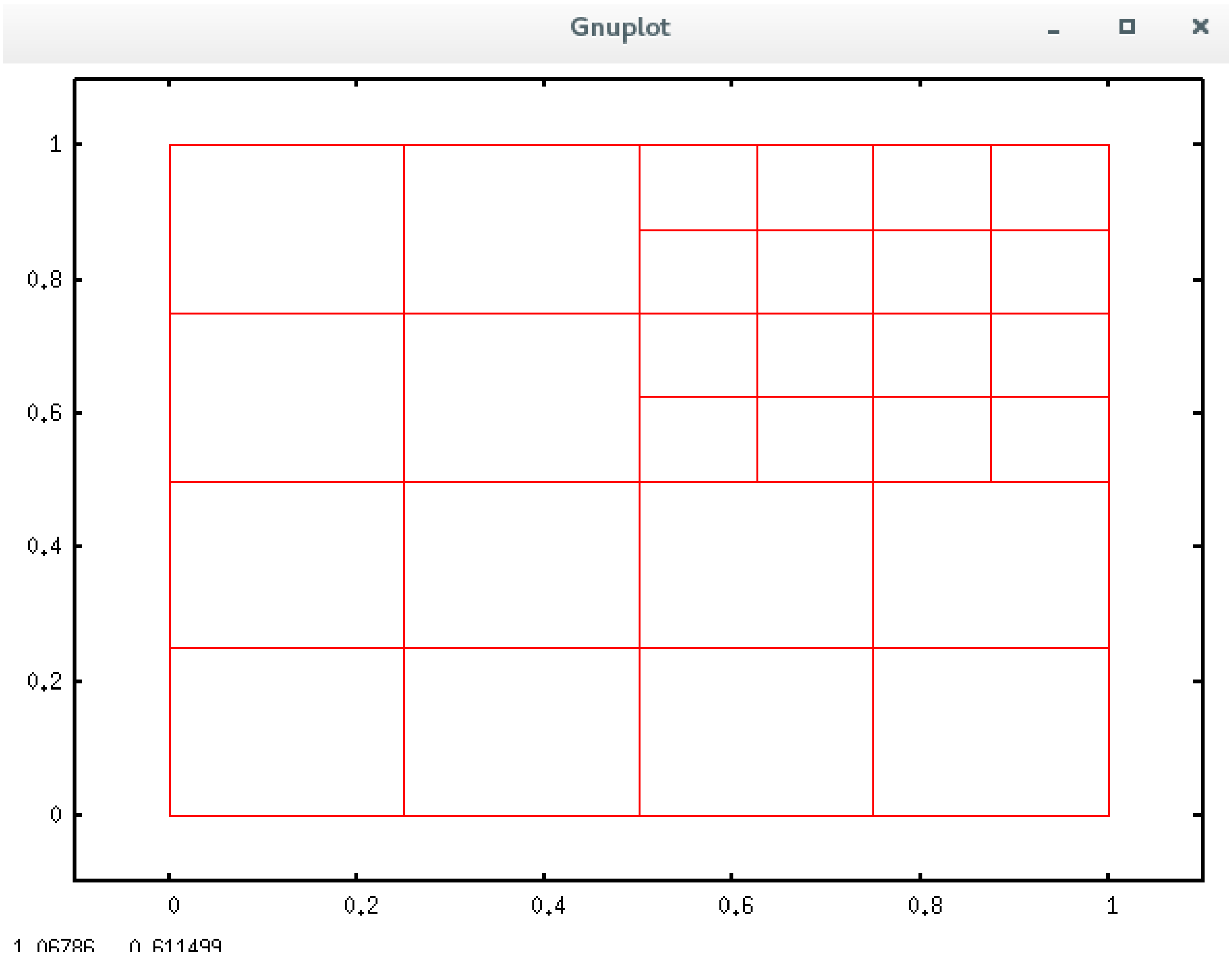
Screen-shots to prove each task and detailed information describing your output to gnuplot

<u>Screen-shot 1:</u>
This shows that the quadtree is only creating seven leaves on the gnuplot.



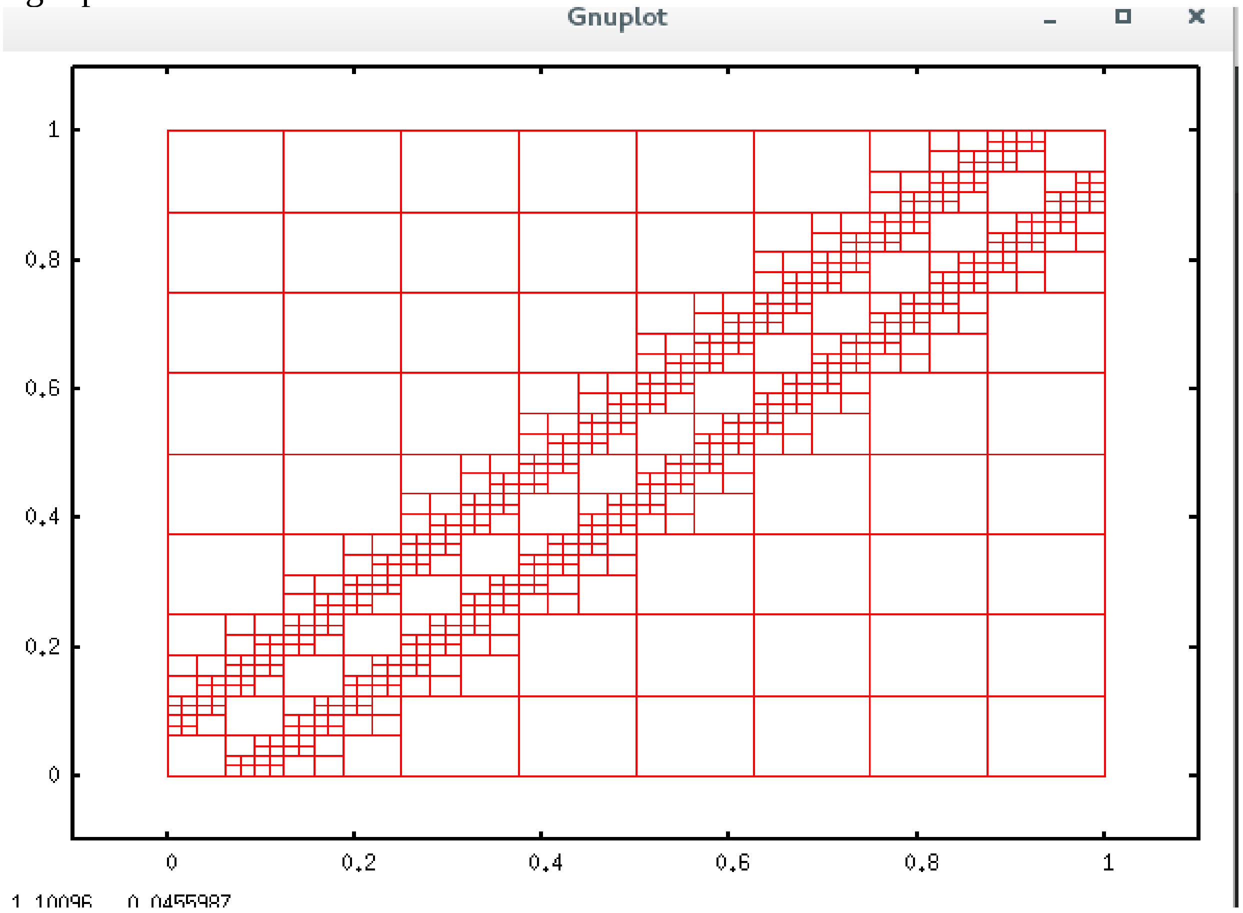
Harshit Verma
ID: 200978548

<u>Screen-shot:2</u>
When I increase my level, it created more leaves and printed out on the gnuplot



Harshit Verma
ID: 200978548

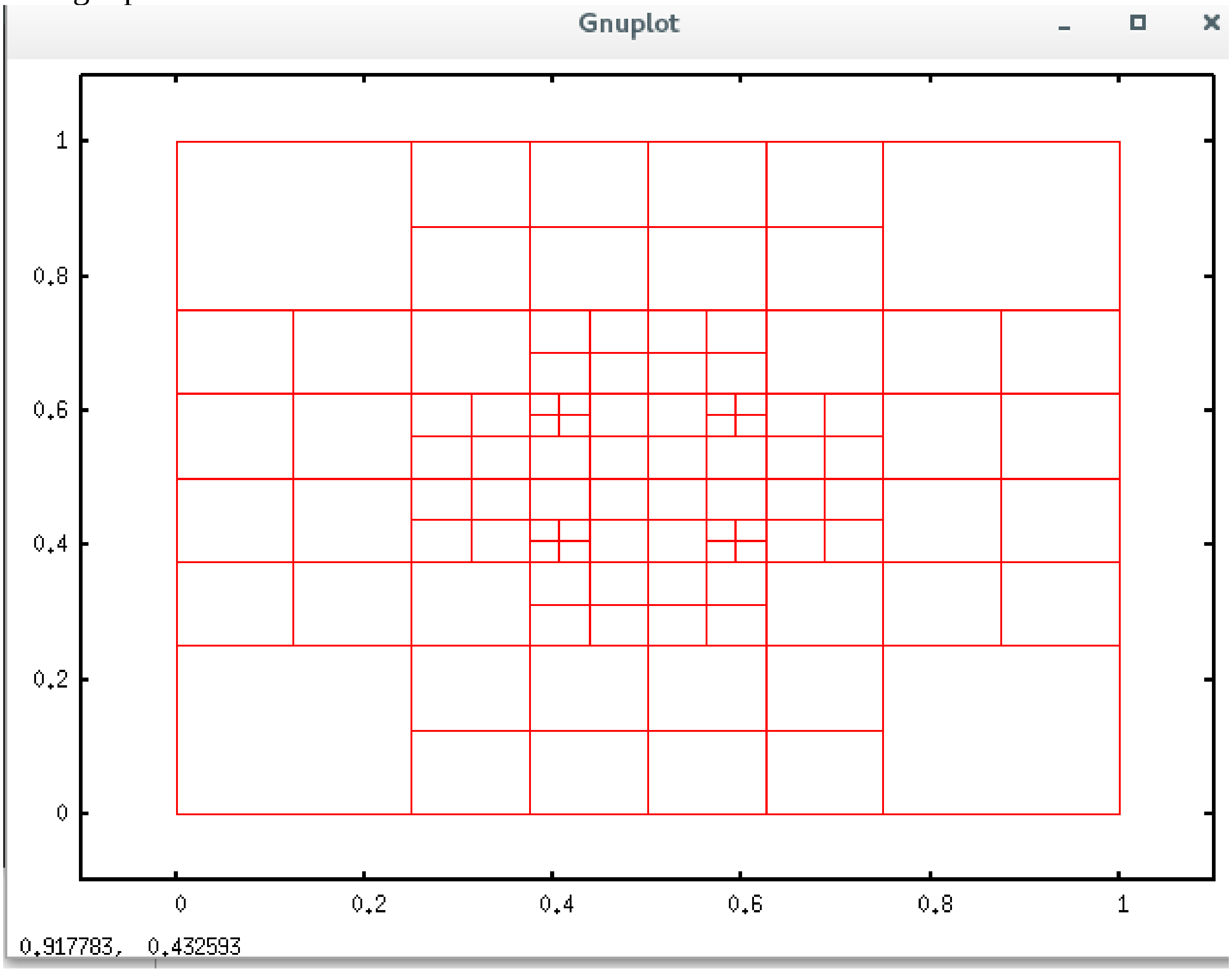
Screen-shot: 3
When I changed my tolerance to 0.8 and choice to 0, it increased the level and created more nodes and leaves.
It shows the last task which is to change the tolerance and see the effects take place on gnuplot.



```
1.10096    0.0455987
```

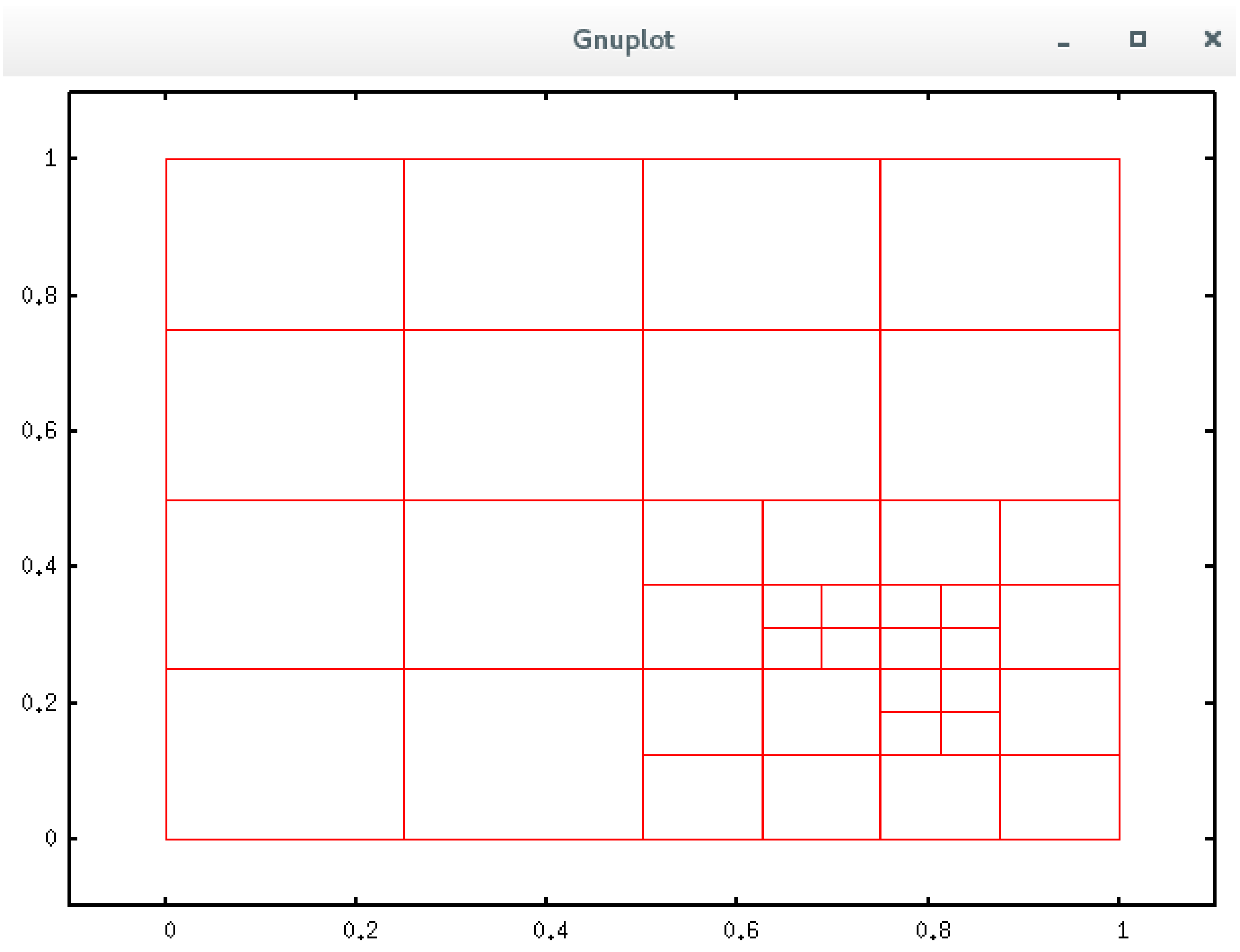
Harshit Verma
ID: 200978548

<u>Screen-shot :4</u>
When I changed my tolerance to 0.8 and choice to 1, it increased the level and created more nodes and leaves.
It shows the last task which is to change the tolerance and see the effects take place on gnuplot.



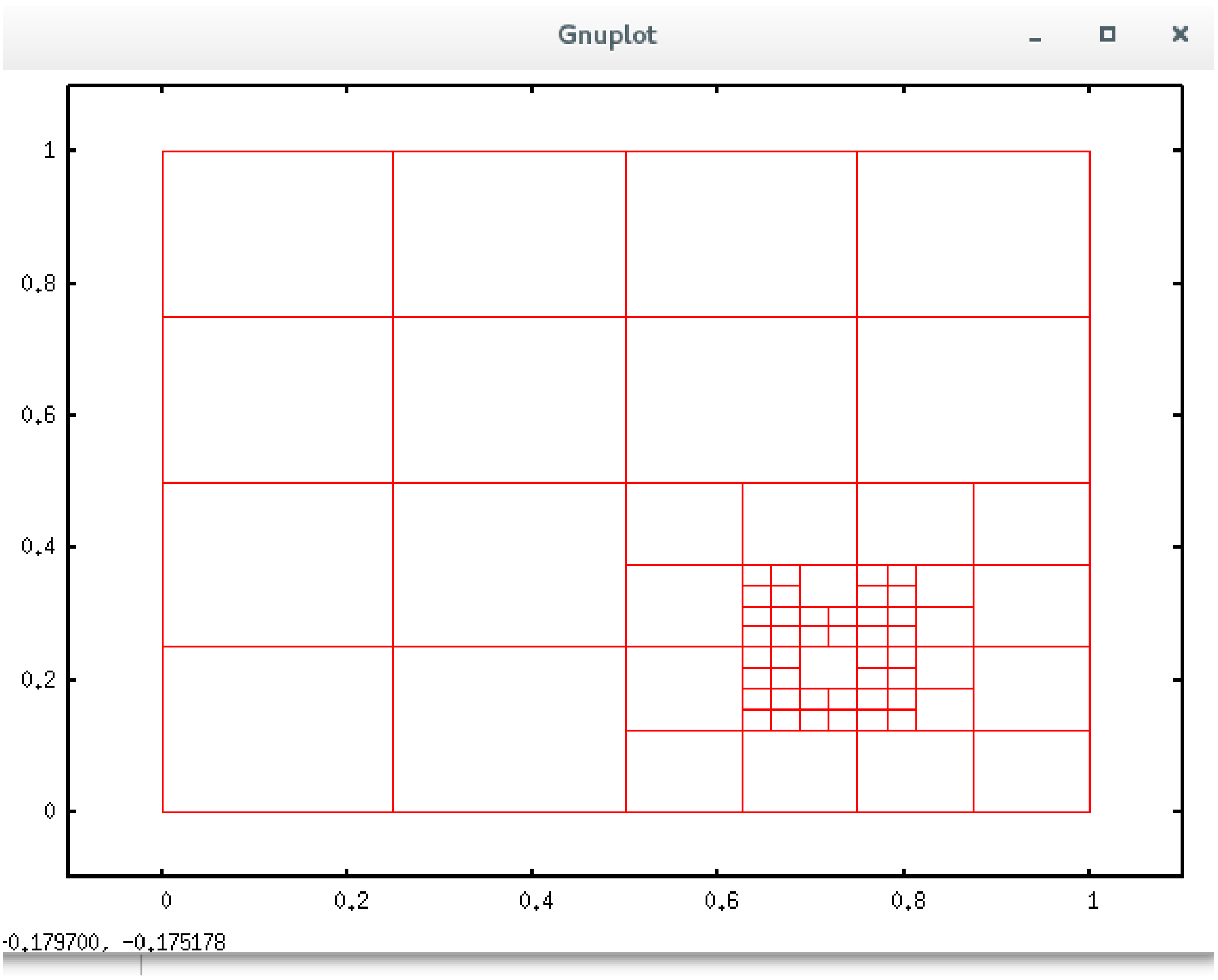
Harshit Verma
ID: 200978548

Screen-shot: 5
When I changed my tolerance to 0.8 and choice to 2, it increased the level and created more nodes and leaves.
It shows the last task which is to change the tolerance and see the effects take place on gnuplot.



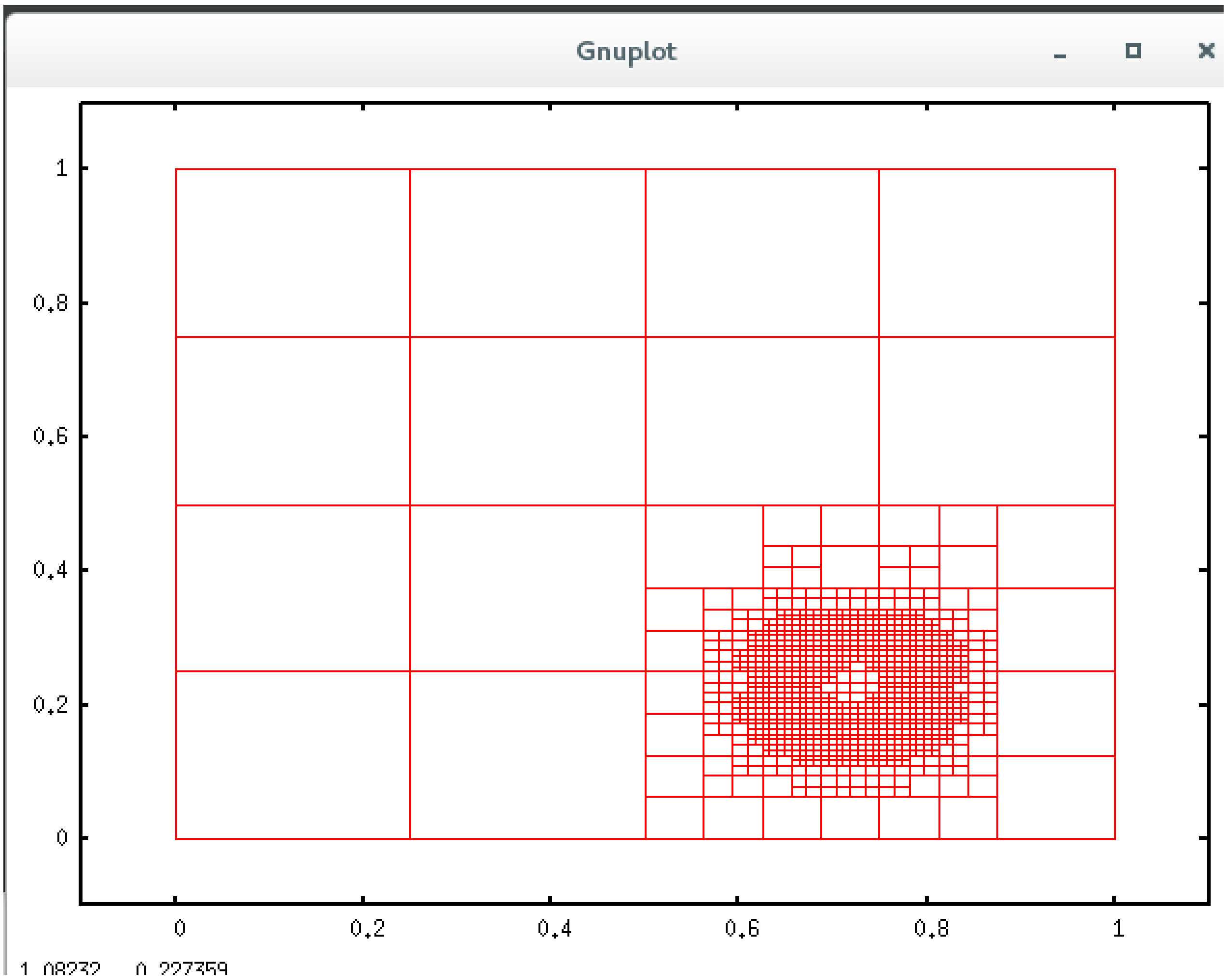
Harshit Verma
ID: 200978548

<u>Screen-shot: 6</u>
When I changed my tolerance to 0.4 and choice to 2, it increased the level and created more nodes and leaves.
It shows the last task which is to change the tolerance and see the effects take place on gnuplot.



Harshit Verma
ID: 200978548

Screen-shot: 7
When I changed my tolerance to 0.05 and choice to 2, it increased the level and created more nodes and leaves.
It shows the last task which is to change the tolerance and see the effects take place on gnuplot.


Gnuplot
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
1.08232 0.227359