KD Trees Implementation

Sergio Rojas- Aguilar

What is a KD Tree

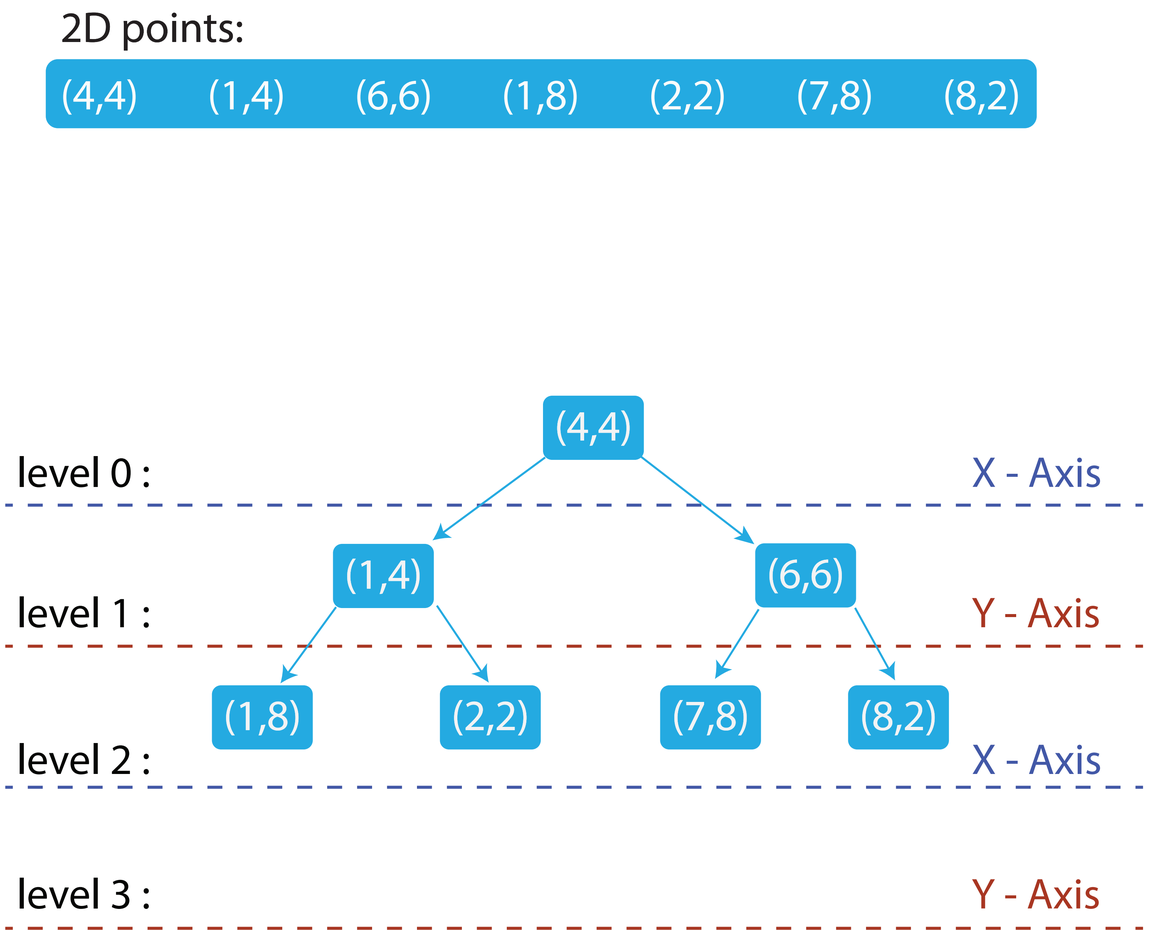
A KD Tree (k-dimensional tree) is a binary search tree used for organizing and searching points in a k-dimensional space. One of its application is performing nearest neighbor searches in k dimensions.

Given N points with dimension K, the tree is organized by cycling through the axes of the dimensions at each level.

What is a KD Tree

A KD Tree (k-dimensional tree) is a binary search tree used for organizing and searching points in a k-dimensional space. One of its application is performing nearest neighbor searches in k dimensions.

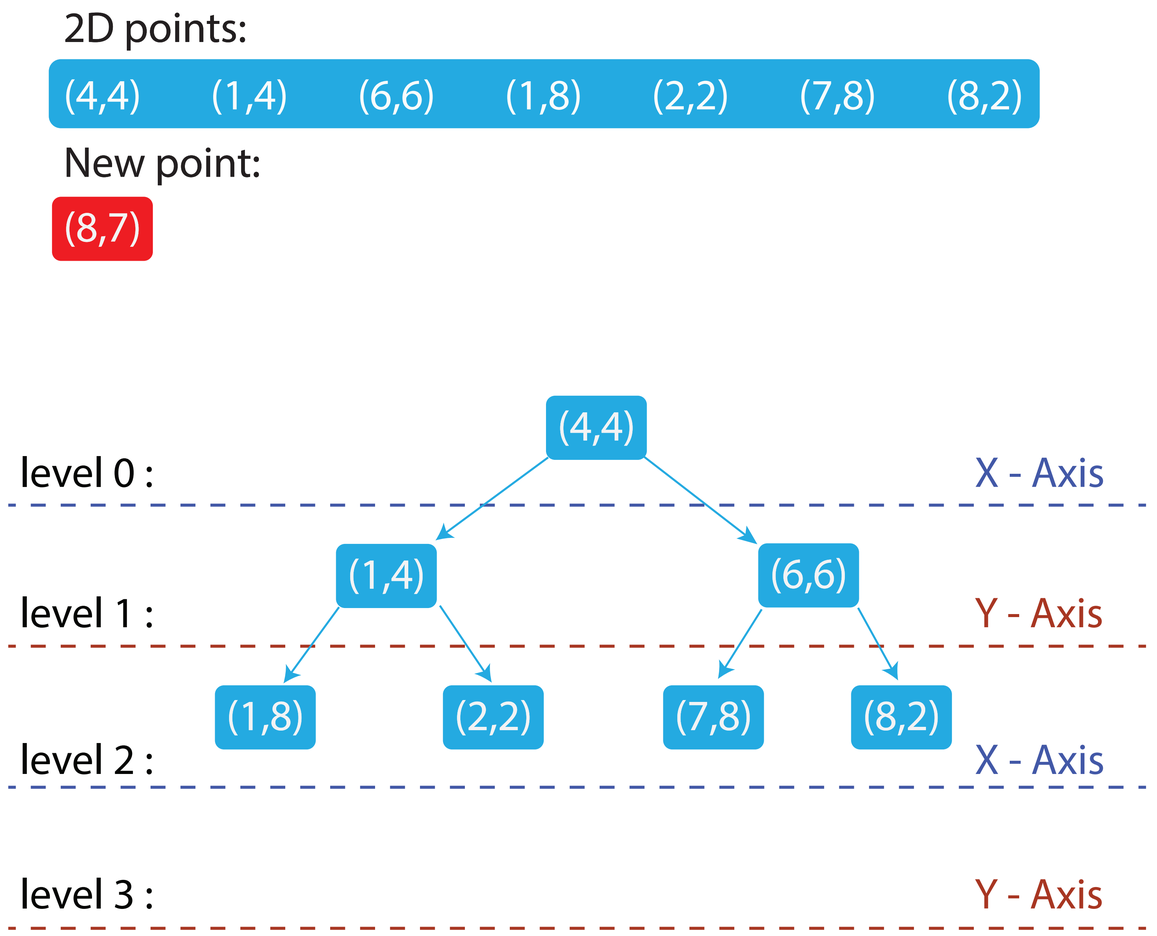
Given N points with dimension K, the tree is organized by cycling through the axes of the dimensions at each level.



What is a KD Tree

A KD Tree (k-dimensional tree) is a binary search tree used for organizing and searching points in a k-dimensional space. One of its application is performing nearest neighbor searches in k dimensions.

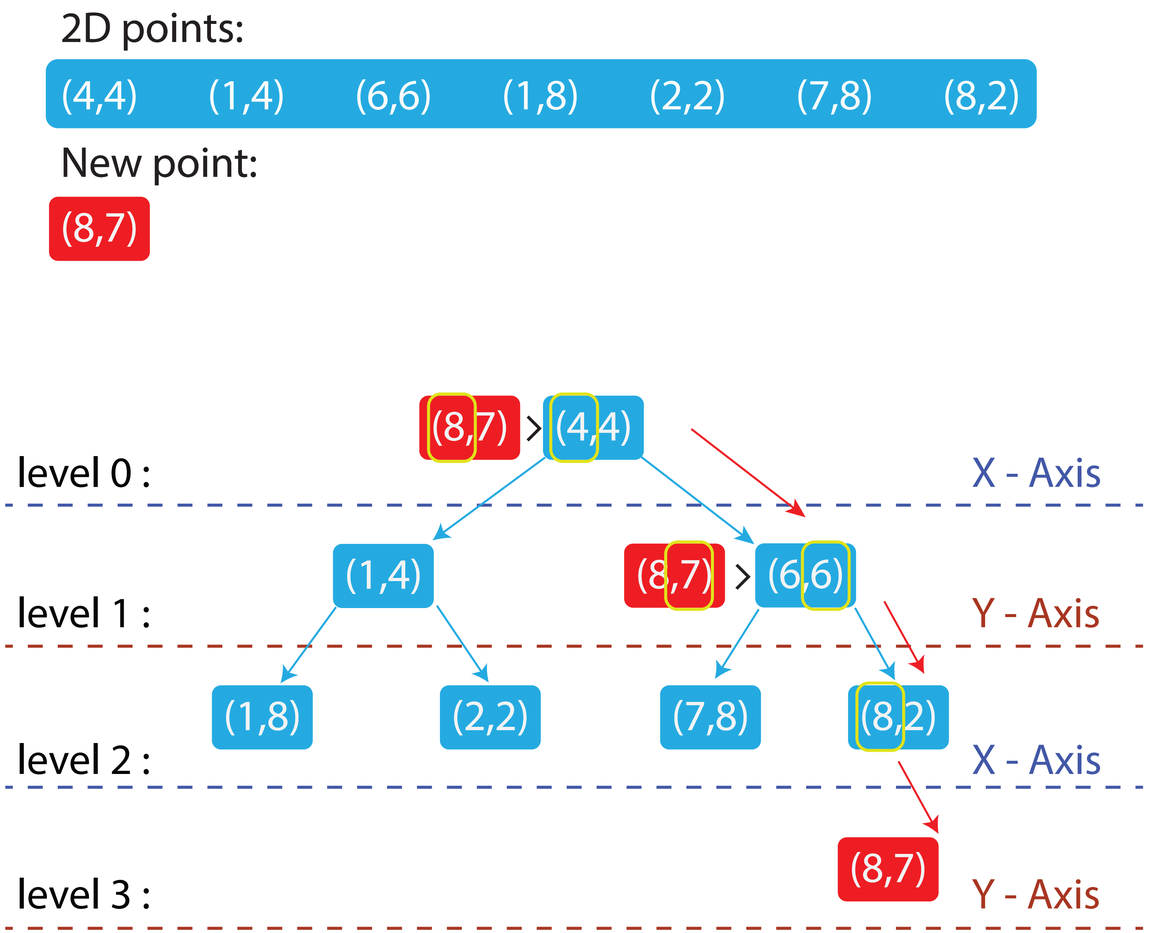
Given N points with dimension K, the tree is organized by cycling through the axes of the dimensions at each level.



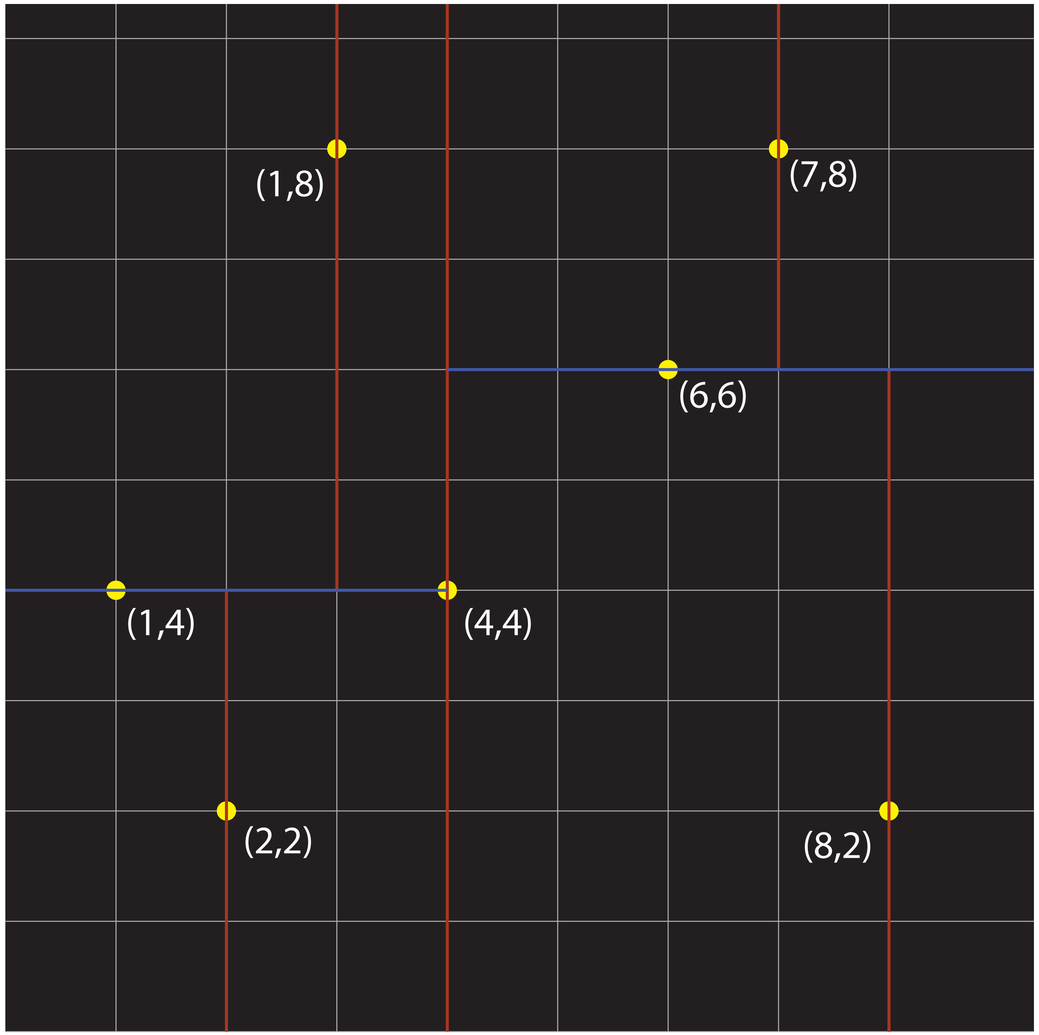
What is a KD Tree

A KD Tree (k-dimensional tree) is a binary search tree used for organizing and searching points in a k-dimensional space. One of its application is performing nearest neighbor searches in k dimensions.

Given N points with dimension K, the tree is organized by cycling through the axes of the dimensions at each level.

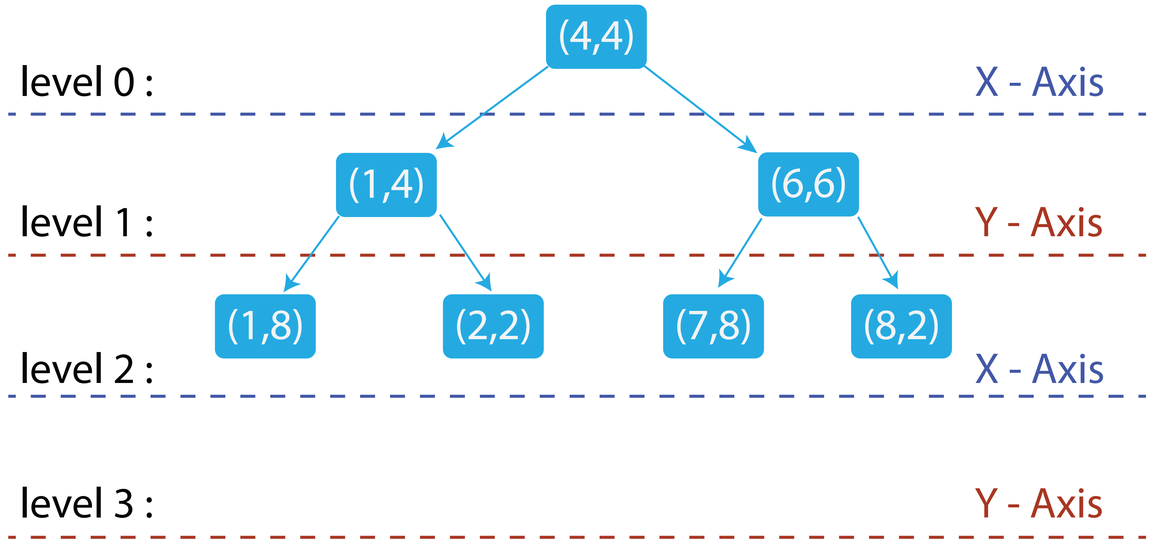


What is a KD Tree

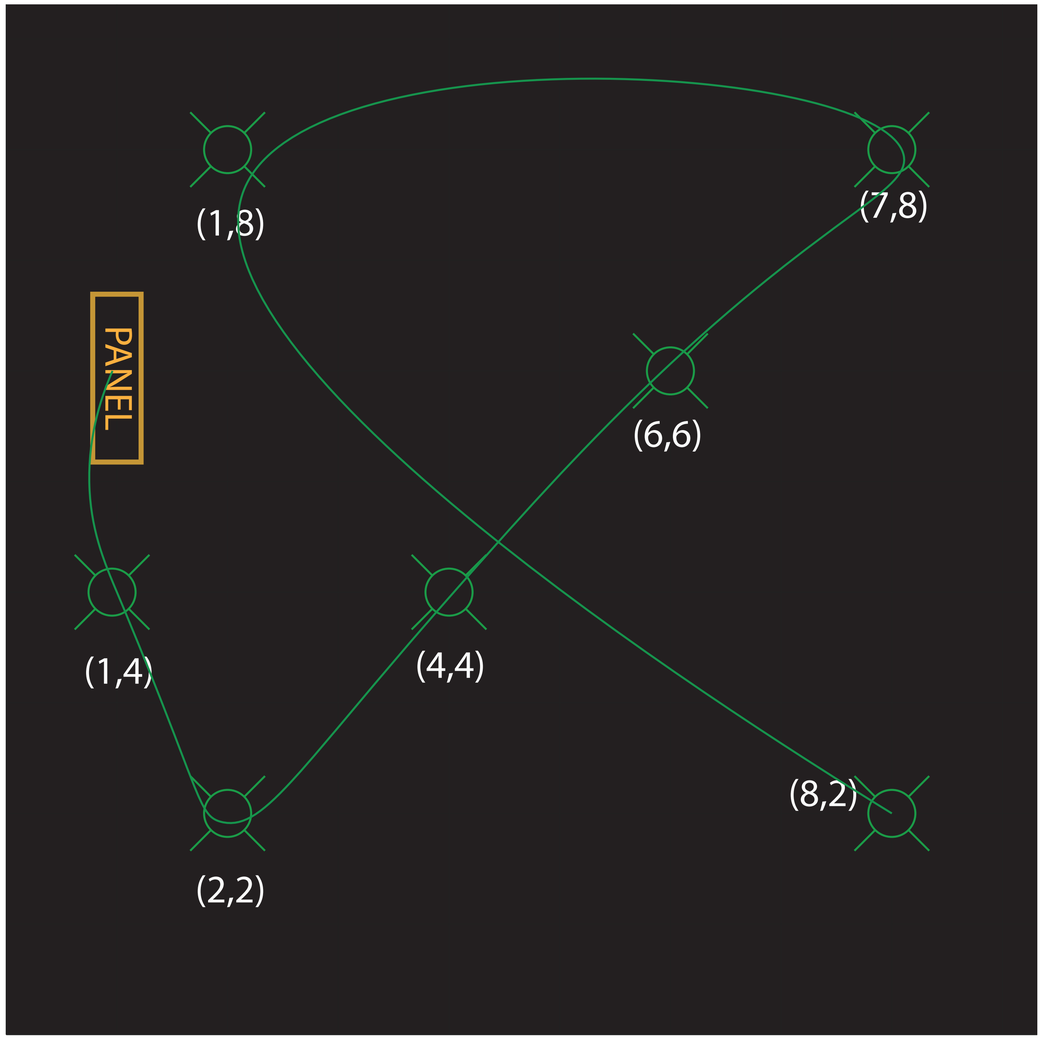


2D points:

(4,4) (1,4) (6,6) (1,8) (2,2) (7,8) (8,2)



Examples



2D points:

(4,4) (1,4) (6,6) (1,8) (2,2) (7,8) (8,2)

Panel Location : (1,6)

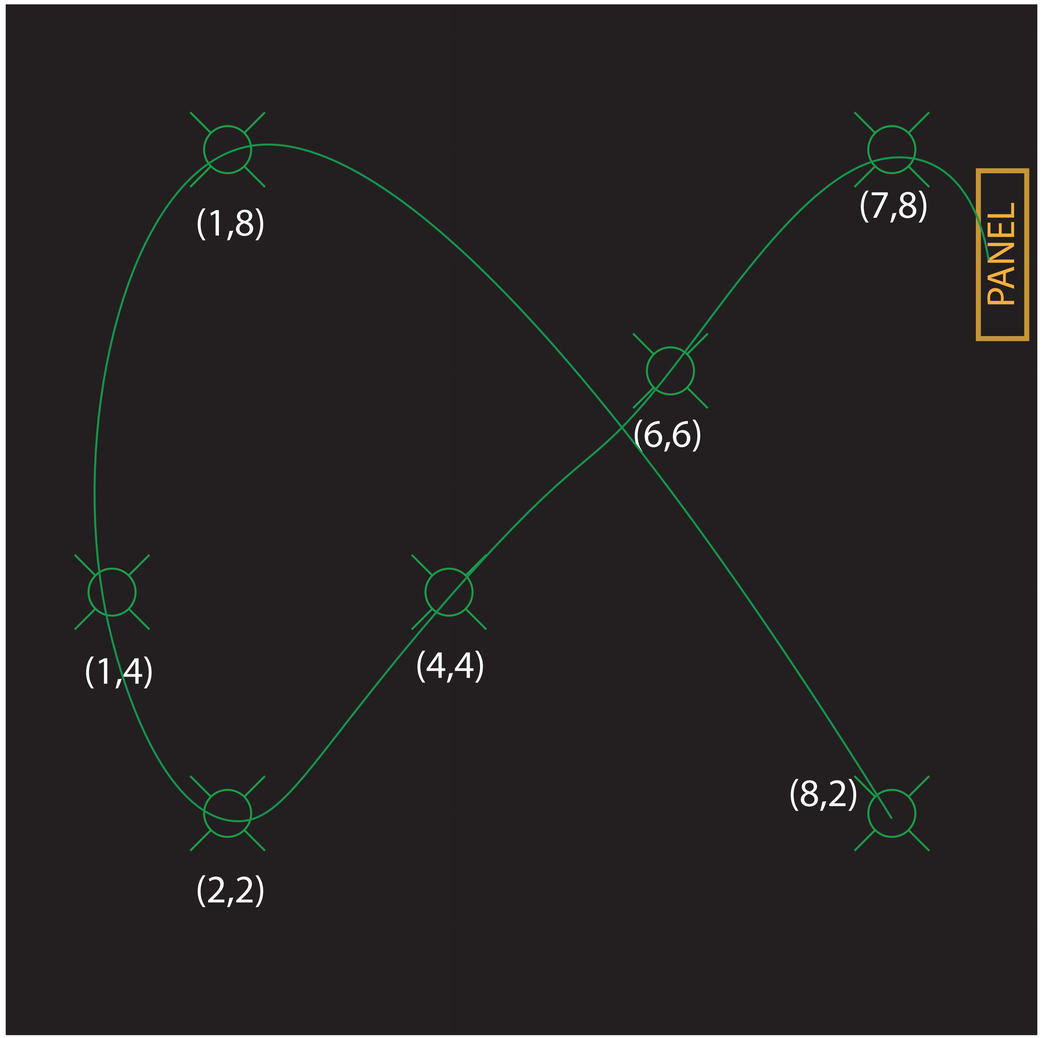
```
sergr@DESKTOP-S95NI0C MINGW64 ~/bootcamp/K-d-Tree-Solution-for-Circuit-Data/app (main)
$ ./main.exe
Please enter relative file location of csv :../testData/PresentationTest.csv

Please enter Panel X axis location :1

Please enter Panel Y axis location :6
new node is new root:4
current left is node the new node :1
current right is node the new node :6
current right is node the new node :1
current left is node the new node :2
Best node is : 6, 6
new node is new root:1
current right is node the new node :7
current left is node the new node :8
new dist: 5.38516   compared to best dist:3.40282e+38
new dist: 2.23607   compared to best dist:5.38516
new dist: 4.47214   compared to best dist:2.23607
end recursion on level 3
end recursion on level 3
new dist: 4.47214   compared to best dist:2.23607
end recursion on level 3
end recursion on level 3
Best node is : 7, 8
new node is new root:1
current right is node the new node :8
new dist: 6   compared to best dist:3.40282e+38
new dist: 6.08276   compared to best dist:6
end recursion on level 2
Best node is : 1, 8
new node is new root:8
new dist: 9.21954   compared to best dist:3.40282e+38
end recursion on level 1
end recursion on level 1
Best node is : 8, 2
(1, 6)(1, 4)(2, 2)(4, 4)(6, 6)(7, 8)(1, 8)(8, 2)

sergr@DESKTOP-S95NI0C MINGW64 ~/bootcamp/K-d-Tree-Solution-for-Circuit-Data/app (main)
$
```

Examples



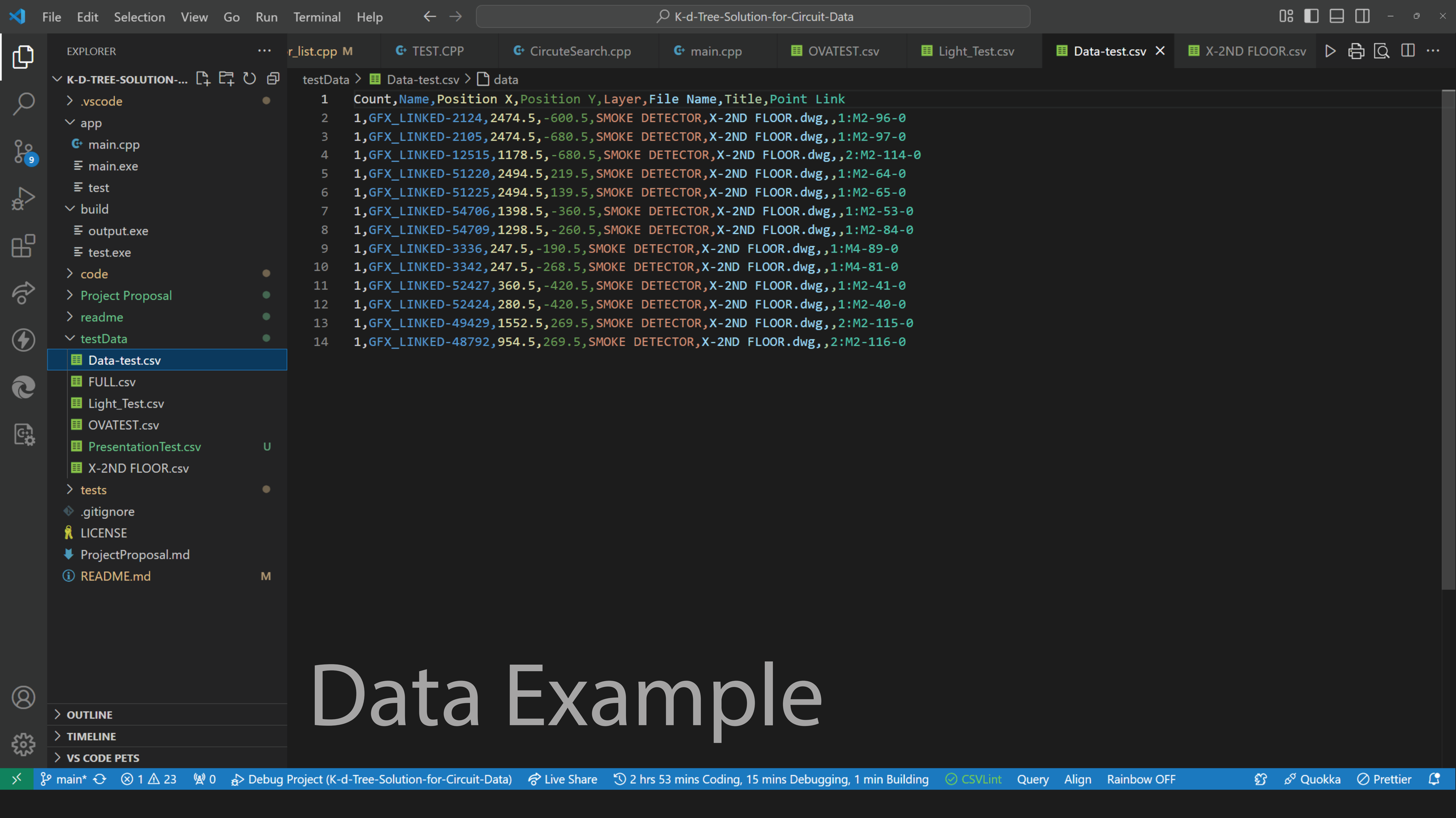
2D points:

(4,4) (1,4) (6,6) (1,8) (2,2) (7,8) (8,2)

Panel Location : (9,7)

```
MINGW64:/c/Users/sergr/boc
new node is new root:1
current right is node the new node :1
current left is node the new node :8
new dist: 2.23607   compared to best dist:3.40282e+38
new dist: 6.08276   compared to best dist:2.23607
new dist: 6   compared to best dist:2.23607
end recursion on level 3
new dist: 6.08276   compared to best dist:2.23607
new dist: 6   compared to best dist:2.23607
end recursion on level 3
Best node is : 1, 4
new node is new root:1
current right is node the new node :8
new dist: 4   compared to best dist:3.40282e+38
new dist: 7.28011   compared to best dist:4
end recursion on level 2
end recursion on level 2
new dist: 7.28011   compared to best dist:4
end recursion on level 2
end recursion on level 2
Best node is : 1, 8
new node is new root:8
new dist: 9.21954   compared to best dist:3.40282e+38
end recursion on level 1
end recursion on level 1
Best node is : 8, 2
(9, 7)(7, 8)(6, 6)(4, 4)(2, 2)(1, 4)(1, 8)(8, 2)

sergr@DESKTOP-S95NI0C MINGW64 ~/bootcamp/K-d-Tree-Solution-for-Circuit-Data/app (main)
$ vvvv
```


File Edit Selection View Go Run Terminal Help
K-d-Tree-Solution-for-Circuit-Data
EXPLORER
K-D-TREE-SOLUTION-...
.vscode
app
main.cpp
main.exe
test
build
output.exe
test.exe
code
Project Proposal
readme
testData
Data-test.csv
FULL.csv
Light_Test.csv
OVATEST.csv
PresentationTest.csv U
X-2ND FLOOR.csv
tests
.gitignore
LICENSE
ProjectProposal.md
README.md M
r_list.cpp M
TEST.CPP
CircuteSearch.cpp
main.cpp
OVATEST.csv
Light_Test.csv
Data-test.csv
X-2ND FLOOR.csv
testData > Data-test.csv > data
1 Count,Name,Position X,Position Y,Layer,File Name,Title,Point Link
2 1,GFX_LINKED-2124,2474.5,-600.5,SMOKE DETECTOR,X-2ND FLOOR.dwg,,1:M2-96-0
3 1,GFX_LINKED-2105,2474.5,-680.5,SMOKE DETECTOR,X-2ND FLOOR.dwg,,1:M2-97-0
4 1,GFX_LINKED-12515,1178.5,-680.5,SMOKE DETECTOR,X-2ND FLOOR.dwg,,2:M2-114-0
5 1,GFX_LINKED-51220,2494.5,219.5,SMOKE DETECTOR,X-2ND FLOOR.dwg,,1:M2-64-0
6 1,GFX_LINKED-51225,2494.5,139.5,SMOKE DETECTOR,X-2ND FLOOR.dwg,,1:M2-65-0
7 1,GFX_LINKED-54706,1398.5,-360.5,SMOKE DETECTOR,X-2ND FLOOR.dwg,,1:M2-53-0
8 1,GFX_LINKED-54709,1298.5,-260.5,SMOKE DETECTOR,X-2ND FLOOR.dwg,,1:M2-84-0
9 1,GFX_LINKED-3336,247.5,-190.5,SMOKE DETECTOR,X-2ND FLOOR.dwg,,1:M4-89-0
10 1,GFX_LINKED-3342,247.5,-268.5,SMOKE DETECTOR,X-2ND FLOOR.dwg,,1:M4-81-0
11 1,GFX_LINKED-52427,360.5,-420.5,SMOKE DETECTOR,X-2ND FLOOR.dwg,,1:M2-41-0
12 1,GFX_LINKED-52424,280.5,-420.5,SMOKE DETECTOR,X-2ND FLOOR.dwg,,1:M2-40-0
13 1,GFX_LINKED-49429,1552.5,269.5,SMOKE DETECTOR,X-2ND FLOOR.dwg,,2:M2-115-0
14 1,GFX_LINKED-48792,954.5,269.5,SMOKE DETECTOR,X-2ND FLOOR.dwg,,2:M2-116-0
Data Example
OUTLINE
TIMELINE
VS CODE PETS
main* 1 23 0 Debug Project (K-d-Tree-Solution-for-Circuit-Data) Live Share 2 hrs 53 mins Coding, 15 mins Debugging, 1 min Building CSVLint Query Align Rainbow OFF Quokka Prettier