



Illinois Institute of Technology

Separating Points by Axis Parallel Lines in Python

Prof. [Gruia Calinescu](#)

Submitted By:

[Sachin Janwalkar](#) A20479201

SEPERATING POINTS BY AXIS PARALLEL LINES

Solution greedy algorithm. In each iteration we select the line which separates maximum points.

Pseudocode:

1. Read input file in ip_data
2. Extract number of points 'n', x-coordinates & y-coordinates from ip_data
3. Sort x_coord and y_coord. # x-coordinate & y-coordinate
4. Create a list of ver_lines and hor_lines by calculating distance between points
5. For i in range(n-1):
6. ver_lines.append(x_coord[i]+x_coord[i+1]/2) #These points represent imaginary lines
7. hor_lines.append(y_coord[i]+y_coord[i+1]/2)
8. Create a list of co_ordinates which contains all indexes for traversal to each point in x_coord and y_coord to every other point.
For i in range(n-1)
 For j in range(i+1,n)
 co_ordinate.append(i,j) # (0,1),(0,2)...(1,2)...(2,3)....so on

Till now we have created vertical and horizontal lines between each pair of x_coord and y_coord and stored it in ver_lines and hor_lines. These lines separate each point in the plane.

Consider values in ver_lines and hor_lines represents a vertical/horizontal line on that particular coordinate respectively. E.g- ver_lines[1] = 2.5, means a vertical line on x=2.5

Time for optimization

1. Initialize selected_line= 0, selected_axis=' '
2. Run till co_ordinate != Null
3. Initialize score=0
4. For v in ver_lines :
5. For c in co_ordinate:
6. If line v separates x_coord[c[0]] and x_coord[c[1]] # Function call
7. Increment score
8. Else:
9. Return False
10. If current score > previous score
11. Update score
12. selected_line= v
13. selected_axis = 'vertical'
14. For h in hor_lines :
15. For c in co_ordinate:
16. If line v separates y_coord[c[0]] and y_coord[c[1]] # Function call
17. Increment score
18. Else:
19. Return False
20. If current score > previous score

```

21.                Update score
22.                selected_line = h
23.                selected_axis = 'horizontal'
24.    If selected_axis = 'vertical'                # storing the result
25.        result.append('v'+selected_line)
26.    Else:
27.        result.append('h'+selected_line)
28.    For c in co_ordinates:                # updating coordinates which separates points
29.        If selected_axis = 'vertical'
30.            If selected_line separates x_coord[c[0]] and x_coord[c[1]]
31.                Remove c from co_ordinate
32.        Else:
33.            If selected_line separates y_coord[c[0]] and y_coord[c[1]]
34.                Remove c from co_ordinate
35.
36. Finally save result in a file.

```

RunTime Analysis:

As written in the pseudocode and the actual program, the optimization code consists of outer WHILE loop which has 2 inner FOR loops(Outer FOR and Inner FOR Loop) .

The WHILE Loop runs till it doesn't become NIL which will takes maximum of 'n' runs for n elements, the outer FOR loop runs for each vertical and horizontal line, so if we have 10 x-coordinated the number of lines require to separate them is (n-1) lines

Therefore, the outer FOR loop runs for (n-1) time

The Inner FOR loop will run for each co-ordinate in the co_ordinate array, so it will run for $n(n+1)/2$ time
 $=n^2$

run time(Optimization) $=O(n*(n-1)*n^2) =O(n^4)$

Process of separating X, Y coordinates takes $O(n)$ and Sorting will take $O(n^2)$ time and Creating co_ordinate array will take $O(n^2)$ time

So, to compute total run time:

Run_time(Seperating X & Y coordinates)+Run_time(Sorting)+Run_time(Creating co_ordinate array) +
 Run_time(optimization i.e. while loop)

$= O(n)+O(n^2) +O(n^2) + O(n^4) = O(n^4)$

How to run the program :

1. Open the project.py file in the zip folder.
2. Run project.py file in pycharm(python)
3. Store the input file in the C:\Users\sachi\PycharmProjects\pythonProject → your python folder directory
4. In the project.py file change the variable instance_no as per the name of input file
For eg. If Input file name instance05 → Initialize instance_no ='05'
5. Run the Program, It will create output file in the same directory with file name
greedy_solution+instance_no+.txt

```
#CS-430 Project
instance_no = '01'      # change the variable as per your instance no: For eg. instance05 ---> change to '05'
ip_filename = "instance"+str(instance_no)+".txt"
```

Working:

- Start by separating the x-coordinates and y-coordinates
- Sort the x and y coordinates
- Form connection between each point in x and y coordinate and store those connection in co_ordinate array, so co_ordinate array contains [(0,1),(0,2)(0,3)...(1,2)..(n-1,n)] such that each point is connected to every other point.
- Create points using distance formula between adjacent x and y coordinate points (this points acts as line separating points)
- Now, for each separating point created (using distance formula) check which point separates the maximum number of x and y coordinates using the co_ordinate array and store its result.
- For each iteration store the separating point which maximizes the result for x and y coordinate. If the result if for x-coordinate means vertical line else horizontal line
- Remove the connections between points in co_ordinate array which gets separated by the selected line after each iteration.
- Repeat this process till all the connection in the co_ordinate array is removed.

Failure Instance:

The algorithm fails to produce output in case the input value given are decimal values

Consider input file as:

```
instance10 - Notepad
File Edit Format View Help
5
7 14
2 3
100 12
9 9.5
6 3|
```

As it can be seen the y-coordinate of the 4th point is in decimal points i.e. 9.5 =, So during separating x and y coordinates it produces an error as.

It's a simple fix just need to change data type of the operation to float and that shall do the trick.

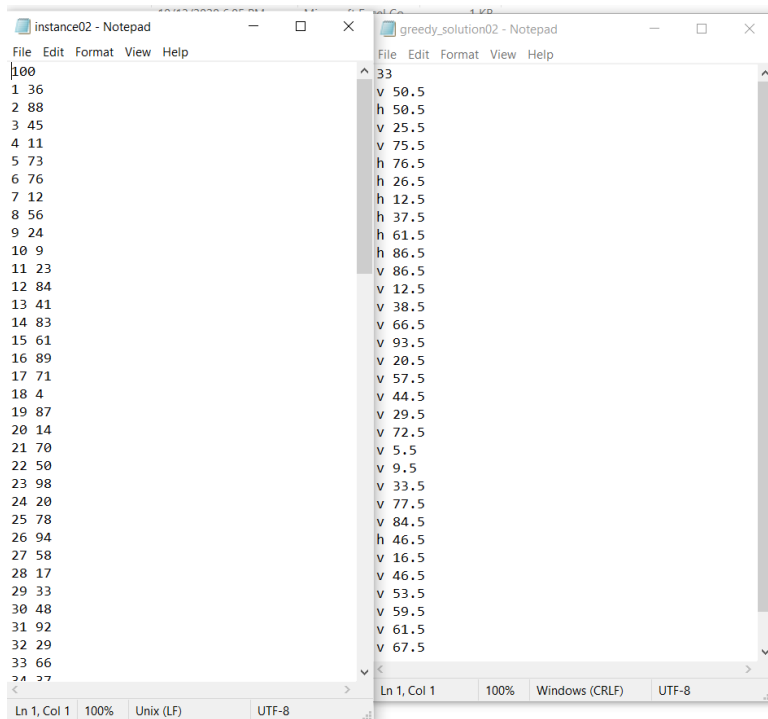
For the rest of the input cases and trial runs that I performed (without inputs containing decimal points), it produced optimum solution for each input instance. As it computes distance between adjacent points using distance formula and finalizes the point which produces the maximum separation during each iteration until all the points are separated, It gives optimum solution in almost all the cases.

Output Screenshots:

› Sachin Janwalkar › PycharmProjects › pythonProject

	Name	Date modified	Type	Size
	Country clusters	10/13/2020 6:05 PM	Microsoft Excel Co...	1 KB
✦	CS-430 Optimum solution	5/10/2021 1:52 PM	Python File	6 KB
✦	CS-430 Project	5/9/2021 11:50 AM	Python File	2 KB
✦	Dictionaries	3/7/2021 3:30 PM	Python File	1 KB
✦	Dummies	10/5/2020 9:26 PM	Microsoft Excel Co...	2 KB
	Example	10/14/2020 7:24 PM	Microsoft Excel Co...	1 KB
	For & While	9/18/2020 7:03 PM	Python File	1 KB
	Function	9/18/2020 7:20 PM	Python File	1 KB
card	greedy_solution00	4/28/2021 6:49 PM	Text Document	1 KB
	greedy_solution01	5/10/2021 12:26 AM	Text Document	1 KB
	greedy_solution02	5/10/2021 12:28 AM	Text Document	1 KB
	greedy_solution03	5/9/2021 12:12 PM	Text Document	1 KB
	greedy_solution04	5/9/2021 12:12 PM	Text Document	1 KB
	greedy_solution05	5/10/2021 12:26 AM	Text Document	1 KB
	greedy_solution06	5/9/2021 11:22 AM	Text Document	1 KB
	greedy_solution10	5/10/2021 1:50 PM	Text Document	1 KB
	Hello World	3/26/2021 2:30 PM	Python File	1 KB
	Import	9/20/2020 7:20 PM	Python File	1 KB
	instance00	4/28/2021 6:48 PM	Text Document	1 KB
	instance01	5/9/2021 12:10 PM	Text Document	1 KB
	instance02	5/9/2021 12:10 PM	Text Document	1 KB
	instance03	5/9/2021 12:11 PM	Text Document	1 KB
	instance04	5/9/2021 12:11 PM	Text Document	1 KB
	instance05	5/9/2021 12:11 PM	Text Document	1 KB
	instance06	5/3/2021 5:45 PM	Text Document	1 KB
	instance10	5/10/2021 1:43 PM	Text Document	1 KB

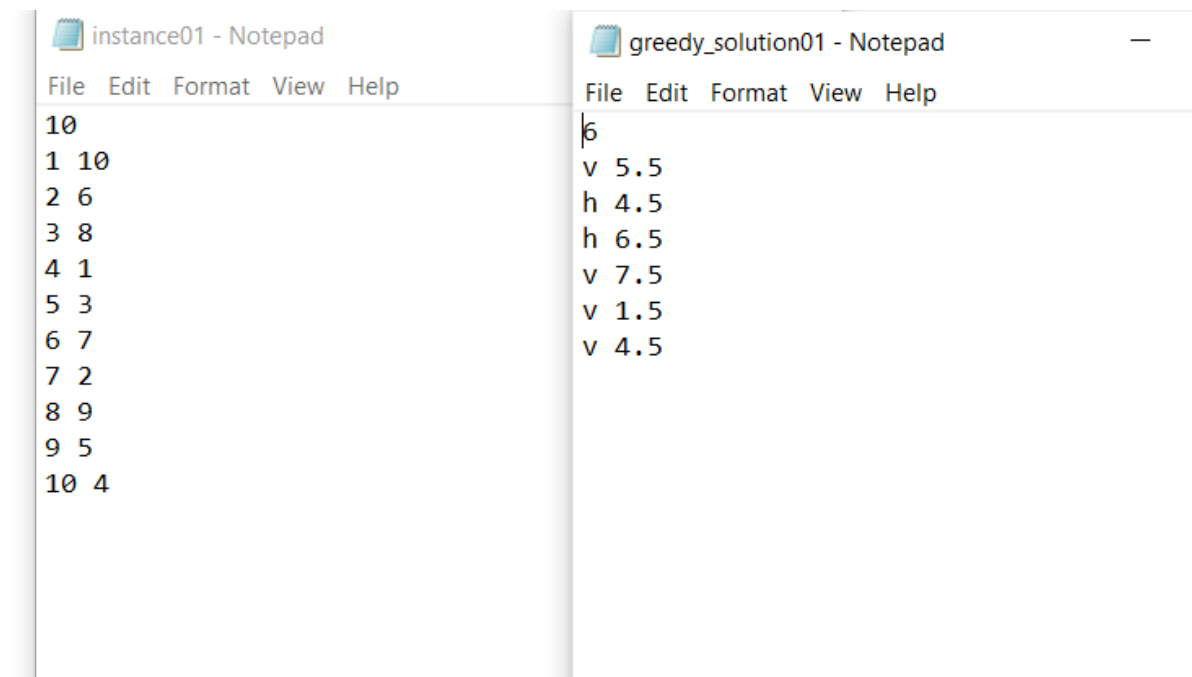
Opening instance02 and greedy_solution02



```
instance02 - Notepad
File Edit Format View Help
100
1 36
2 88
3 45
4 11
5 73
6 76
7 12
8 56
9 24
10 9
11 23
12 84
13 41
14 83
15 61
16 89
17 71
18 4
19 87
20 14
21 70
22 50
23 98
24 20
25 78
26 94
27 58
28 17
29 33
30 48
31 92
32 29
33 66
34 77

greedy_solution02 - Notepad
File Edit Format View Help
33
v 50.5
h 50.5
v 25.5
v 75.5
h 76.5
h 26.5
h 12.5
h 37.5
h 61.5
h 86.5
v 86.5
v 12.5
v 38.5
v 66.5
v 93.5
v 20.5
v 57.5
v 44.5
v 29.5
v 72.5
v 5.5
v 9.5
v 33.5
v 77.5
v 84.5
h 46.5
v 16.5
v 46.5
v 53.5
v 59.5
v 61.5
v 67.5
```

Opening instance01 and greedy_solution01



```
instance01 - Notepad
File Edit Format View Help
10
1 10
2 6
3 8
4 1
5 3
6 7
7 2
8 9
9 5
10 4

greedy_solution01 - Notepad
File Edit Format View Help
6
v 5.5
h 4.5
h 6.5
v 7.5
v 1.5
v 4.5
```