

Puzzle Input:

⇒ Lines of vents in the format
 $x_1, y_1 \rightarrow x_2, y_2$

Task:

⇒ Determine the # of points
where at least two lines overlap

Puzzle Output:

⇒ # of points where at least
two lines overlap

Possible Programmable Solutions:

⇒ Point object contains x, y coords. When
one appears, place in one ~~array~~/dict.
If it appears again, remove from
~~array~~/dict and increment counter

⇒ For dict, I will use closed addressing
where the point will map to the
product of its x and y vals

⇒ Path object contains two point
objects (start/end point)

⇒ For the closed addressing
schema, we will use LinkedLists
with add to front and move back
functionality. ~~This way, once a point~~
~~occurs more than once, we can move it~~
~~to the back of the list. If the list~~
~~hits a point with ≥ 2 occurrences,~~
~~we know the point...~~ now that I'm
writing it out this isn't necessary.
We need to confirm the point has
been seen ≥ 2 x so we do not re-add
it to the list again. Add to front
will still be used since it's $O(1)$.

	-	1	2	3	4	5	6	7	8	9
0										1
1										1
2		1	1							1
3				1						1
4				1						1
5				1						1
6				1						1
7	1	1	1	1	1					1
8	1	1	1	1	1	1	1	1	1	1
9					1					

Issue: Not considering diagonal lines

Part 2:

⇒ Now considering
diagonals

⇒ Need to consider
direction of diagonals