--- Day 15: Beacon Exclusion Zone ---

You feel the ground rumble again as the distress signal leads you to a large network of subterranean tunnels. You don't have time to search them all, but you don't need to: your pack contains a set of deployable sensors that you imagine were originally built to locate lost Elves.

The sensors aren't very powerful, but that's okay; your handheld device indicates that you're close enough to the source of the distress signal to use them. You pull the emergency sensor system out of your pack, hit the big button on top, and the sensors zoom off down the tunnels.

Once a sensor finds a spot it thinks will give it a good reading, it attaches itself to a hard surface and begins monitoring for the nearest signal source beacon. Sensors and beacons always exist at integer coordinates. Each sensor knows its own position and can determine the position of a beacon precisely; however, sensors can only lock on to the one beacon closest to the sensor as measured by the Manhattan distance. (There is never a tie where two beacons are the same distance to a sensor.)

It doesn't take long for the sensors to report back their positions and closest beacons (your puzzle input). For example:

```
Sensor at x=2, y=18: closest beacon is at x=-2, y=15
Sensor at x=9, y=16: closest beacon is at x=10, y=16
Sensor at x=13, y=2: closest beacon is at x=15, y=3
Sensor at x=12, y=14: closest beacon is at x=10, y=16
Sensor at x=10, y=20: closest beacon is at x=10, y=16
Sensor at x=14, y=17: closest beacon is at x=10, y=16
Sensor at x=8, y=7: closest beacon is at x=2, y=10
Sensor at x=2, y=0: closest beacon is at x=2, y=10
Sensor at x=0, y=11: closest beacon is at x=2, y=10
Sensor at x=20, y=14: closest beacon is at x=25, y=17
Sensor at x=17, y=20: closest beacon is at x=21, y=22
Sensor at x=16, y=7: closest beacon is at x=15, y=3
Sensor at x=14, y=3: closest beacon is at x=15, y=3
Sensor at x=20, y=1: closest beacon is at x=15, y=3
```

So, consider the sensor at `2,18`; the closest beacon to it is at `-2,15`. For the sensor at `9,16`, the closest beacon to it is at `10,16`.

Drawing sensors as `S` and beacons as `B`, the above arrangement of sensors and beacons looks like this:

```
               1    1    2    2
     0    5    0    5    0    5
 0 ....S.......................
 1 ......................S.....
 2 ...............S............
 3 ................SB..........
 4 ............................
 5 ............................
 6 ............................
 7 ..........S.......S.........
 8 ............................
 9 ............................
10 ....B.......................
11 ..S.........................
12 ............................
13 ............................
14 ..............S.......S.....
15 B...........................
16 ...........SB...............
17 ................S..........B
18 ....S.......................
19 ............................
20 ............S......S........
21 ............................
22 .......................B....
```

This isn't necessarily a comprehensive map of all beacons in the area, though. Because each sensor only identifies its closest beacon, if a sensor detects a beacon, you know there are no other beacons that close or closer to that sensor. There could still be beacons that just happen to not be the closest beacon to any sensor. Consider the sensor at `8,7`:

```
               1    1    2    2
     0    5    0    5    0    5
-2 ..........#.................
-1 .........###................
 0 ....S...#####...............
 1 .......#######........S.....
 2 ......#########S............
 3 .....###########SB..........
 4 ....#############...........
 5 ...###############..........
 6 ..#################.........
 7 .#########S#######S#........
 8 ..#################.........
 9 ...###############..........
10 ....B############...........
11 ..S..###########............
12 ......#########.............
13 .......#######..............
14 ........#####.S.......S.....
15 B........###................
16 ..........#SB...............
17 ................S..........B
18 ....S.......................
19 ............................
20 ............S......S........
21 ............................
22 .......................B....
```

This sensor's closest beacon is at `2,10`, and so you know there are no beacons that close or closer (in any positions marked `#`).

None of the detected beacons seem to be producing the distress signal, so you'll need to work out where the distress beacon is by working out where it isn't. For now, keep things simple by counting the positions where a beacon cannot possibly be along just a single row.

So, suppose you have an arrangement of beacons and sensors like in the example above and, just in the row where `y=10`, you'd like to count the number of positions a beacon cannot possibly exist. The coverage from all sensors near that row looks like this:

```
                 1    1    2    2
       0    5    0    5    0    5
 9 ...#########################...
10 ..####B######################..
11 .###S#############.###########.
```

In this example, in the row where `y=10`, there are `26` positions where a beacon cannot be present.

Consult the report from the sensors you just deployed. In the row where `y=2000000`, how many positions cannot contain a beacon?

To begin, get your puzzle input.

Answer: [Submit]

You can also [Share] this puzzle.