



## Test Data for Seeing the Boundary

### Algorithms:

Note that there are many ways to get the computational geometry wrong. In addition, one must be very careful to avoid rounding mistakes in floating point numbers. Our low scoring solutions make errors of various sorts, and are not covered below.

Let  $d$  be the maximum number of vertices in any rock.

*Algorithm 1:* Consider the fence as a circular list. Each rock obscures a range of fence posts. Determine the range for each fence post by sorting the angles of the rocks from Farmer Don's location and find the right-most and left-most rock obscured (being careful to deal with wrap-around). Collect these ranges. After processing all rocks, sort the ranges and determine loop through the ranges, counting any fence post not covered by a range.

This algorithm takes  $O(R \log R + N \log d)$  time. It is expected to receive full marks

*Algorithm 2:* Keep a binary array of fence posts. For each rock, determine the range obscured by the rock and reset those data points.

This algorithm takes  $O(N R)$  time. It is expected to receive about 80% of the points.

*Algorithm 3:* For each rock and fence post, run a ray to determine if the rock obscures the fence posts.

This algorithm takes  $O(N R)$  time. It is expected to receive about 72% of the points.



---

Test Data

Test #	Points	N	R	Answer
1	4	100	1	220
2	4	100	1	220
3	4	100	1	218
4	4	100	1	246
5	4	100	1	204
6	4	100	1	269
7	4	40	1	144
8	4	40	1	144
9	4	1500	100	2257
10	4	4000	1500	13
11	4	5000	1800	27
12	4	6000	2000	38
13	4	8000	2400	8244
14	4	10000	3000	4
15	4	16000	3700	0
16	4	28000	5000	108
17	4	40000	6400	288
18	4	30000	373	88507
19	4	62000	9000	83
20	4	91000	12000	368
21	4	123000	15000	43
22	4	164000	17000	20
23	4	250000	23000	2663
24	4	340000	23000	28
25	4	500000	30000	2518