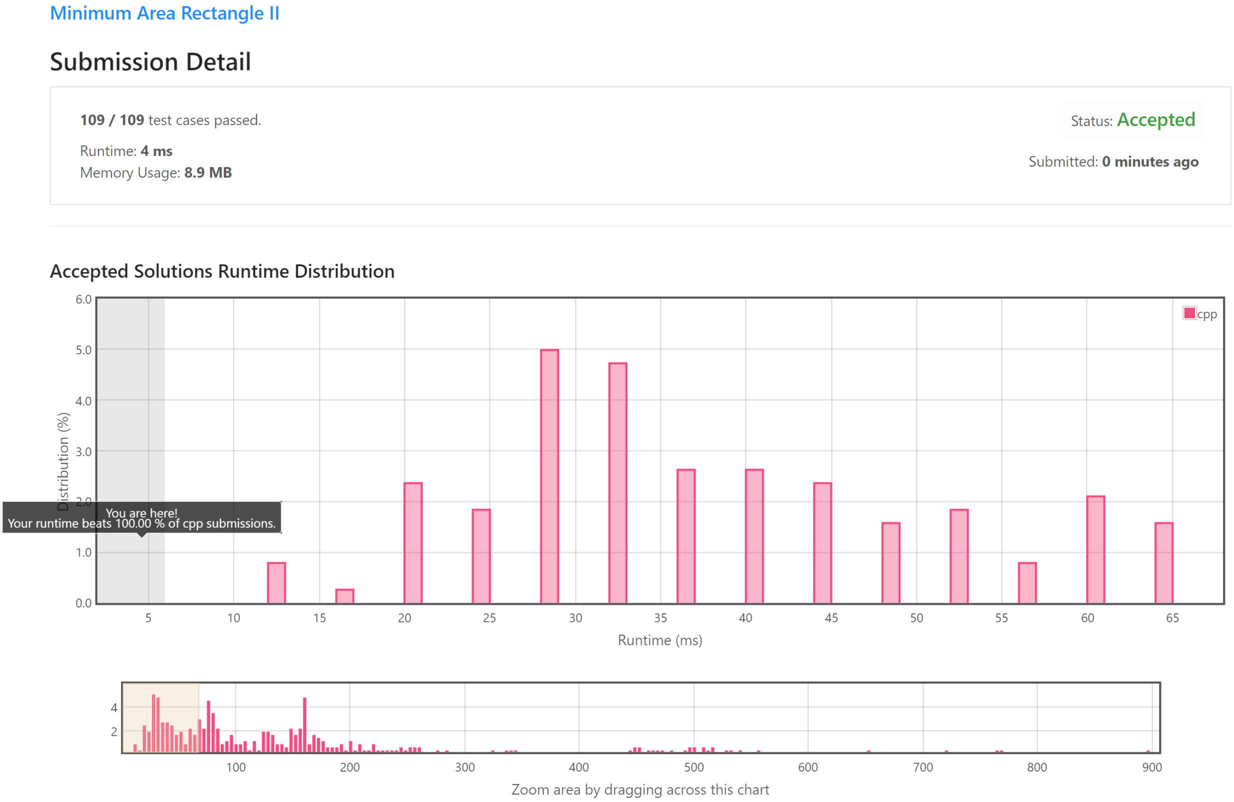
Find two parallel edges using hashing. For an edge pq, use its slope and the y-coordinate of the intersection point between the lines pq' and $x = 0$ during hashing, where pq' is perpendicular to pq. Easy to find the minimum area one by enumerating the points in sorted order. $O(n^2)$.

Remark. 1. Some algorithms in the discussion forum use hashing to find two diagonals that have equal length and share the same middle point, then they enumerate all possible rectangles to find the one with minimum area, but the total number of rectangles is $\Theta(n^2 \log n)$ [1]. (the $O(n^2 \log n)$ upper bound is by [Sharir'89, Personal communication] in the paper)
2. In this problem, all input coordinates are integers, so the above approach gives $n^2 \cdot 2^{O(\log U/\log\log U)}$ running time (not optimal), because consider a group of pairs of points with the same center and distance to the center, the squared distance can be written as the form $L^2 = a^2 + b^2$, where a and b are integers. Any two pairs in the same group can form a rectangle. Using the sum of squares function, each group has size $O(\sigma(U)) = 2^{O(\log U/\log\log U)}$.



Minimum Area Rectangle II

Submission Detail

109 / 109 test cases passed.

Runtime: **4 ms**

Memory Usage: **8.9 MB**

Status: Accepted

Submitted: **0 minutes ago**

Accepted Solutions Runtime Distribution

You are here!
Your runtime beats 100.00 % of cpp submissions.

References

[1] Marc J Van Kreveld and Mark T De Berg. Finding squares and rectangles in sets of points. *BIT Numerical Mathematics*, 31(2):202–219, 1991.