

Analysis: Staying Hydrated

The [Manhattan distance](#) between a point $P = (x, y)$ and an axis-parallel rectangle R with the lower-left corner (x_1, y_1) and the upper-right corner (x_2, y_2) is $d(P, R) = \max(x_1 - x, x - x_2, 0) + \max(y_1 - y, y - y_2, 0)$. Intuitively, the first term represents the minimal horizontal movement from P to the nearest point inside the x -interval $[x_1, x_2]$, which could be 0 if P itself is in that interval. Similarly, the second term represents the minimal vertical movement from P to the nearest point inside the y -interval $[y_1, y_2]$.

Test Set 1

In this test set, the furnished area of the room is so small that we can afford testing every point $P = (x, y)$ in the range $-100 \leq x, y \leq 100$ and select the point with the smallest total distance to all rectangular objects. The time complexity of such a brute-force algorithm is $O(\mathbf{K}WH)$, where \mathbf{K} is the number of input objects and W and H is the width and the height of the smallest axis-parallel rectangle covering all input rectangles. Note that it is not necessary to test any points outside this covering rectangle as moving from such a point towards the covering rectangle would reduce the distance to all input rectangular objects. The time complexity can be further improved to $O(\mathbf{K}(W + H))$ if we realize that the optimal x - and y -coordinates are in fact independent and can be calculated separately.

Test Set 2

In this test set, the furnished area of the room may be huge, therefore, even the improved version of the brute-force algorithm would be too slow. However, with a little bit of sorting, we can find the optimal location of the bottle without ever computing any distances.

Let us consider finding the optimal x -coordinate of the bottle (the optimal y -coordinate can be found in the same way). Imagine that we are seeking the optimal location by sweeping the coordinate plane from left to right. Suppose we are currently at the position x and let $a(x)$ be the number of rectangles that are strictly ahead of x , i.e. $x < x_1$, where x_1 is the left coordinate of the rectangle. Similarly, let $b(x)$ be the number of rectangles that are behind x in the non-strict sense, namely, $x_2 \leq x$. Now, if $a(x) > b(x)$, then x is not the optimal location as moving one step to the right would reduce the total horizontal distance to the rectangles by $a(x) - b(x) > 0$.

What happens to the value of $a(x) - b(x)$ as we sweep the plane from left to right? For a sufficiently small x , which is strictly to the left of all rectangles, $a(x) = \mathbf{K}$ and $b(x) = 0$, and therefore, $a(x) - b(x) = \mathbf{K} > 0$. Conversely, for a sufficiently large x , it is the other way around and $a(x) - b(x) = -\mathbf{K} < 0$. And since $a(x)$ is a decreasing function while $b(x)$ is an increasing function, the difference $a(x) - b(x)$ is also a decreasing function. What does it mean for our plane sweeping approach? As long as $a(x) - b(x) > 0$, we should keep moving to the right as, by doing so, we are reducing the total distance to the rectangles. But as soon as $a(x) - b(x) \leq 0$, we have found the optimal x -coordinate, since further reduction of the total distance is not possible.

In summary, we have reduced the task of calculating the optimal Manhattan distance to finding the smallest x with $a(x) - b(x) \leq 0$. This can be done by iterating through the left and right coordinates of rectangles in sorted non-decreasing order and maintaining the values of functions $a(x)$ and $b(x)$. The time complexity of this algorithm is dominated by the sorting, and is therefore $O(\mathbf{K} \log \mathbf{K})$.

Left as an exercise to the reader: This problem can also be solved in $O(\mathbf{K})$ by using [linear-time selection algorithms](#). Though both our test sets should pass with the above $O(\mathbf{K} \log \mathbf{K})$ approach.