Code Jam 2009 - World Finals

Analysis: Lights

There are various ways to solve this problem. The solutions can be split into two kinds of approaches. One way is to solve the problem exactly (or, to arbitrary precision), the other is to approximate the answer with high enough precision. The exact solutions generally try to divide the square into subareas of the same color, compute the area of each subarea separately, and add up the totals for each color. We discuss these first.

Intersection of (mostly) triangles

First, consider just one light. We want to compute the total area illuminated by the light. To do that, compute the tangent lines from the light to each pillar, and lines from the light to each corner of the room, and consider each "cone" between two adjacent lines separately. Each cone will either end up hitting a wall, or hitting a pillar. In the first case we get a triangle, whose area we can easily compute. In the second case we get a "quasi-triangle", that is, a triangle minus a disk segment. Here, we need to subtract the area of the disk segment. We can compute the area of a disk segment by subtracting a triangle from the area of a disk sector (a "pie slice").

Once we have the total area covered by each light, we need one more thing: the area covered by both lights. We can take each pair of triangles or quasi-triangles generated in the previous step, and compute the common area between them. Now we need to compute the area of the intersection of two triangles or quasi-triangles.

A simple way to approach this is to first treat quasi-triangles as triangles (include the disk segment). Now we compute the intersection between the two triangles, which gives us a polygon (up to six sides). If one or both of the triangles were actual triangles, or when the pillars subtracted from the two quasi-triangles were different, the polygon is the correct answer - there is no need to account for the subtracted disk segments, because each segment is outside the other triangle anyway.

The only nasty case comes up when we have two quasi-triangles ending at the same pillar. In that case, we first compute the intersection polygon, and then we subtract the pillar from the polygon. To do that, remove those edges and parts of edges of the polygon that fall inside the circle and replace them with one edge. The answer will be the area of the reduced polygon, again minus a disk sector cut off by a line, which we already know how to compute.

Line sweeping

Line sweeping is a common technique in computational geometry. We sweep a vertical line from the left edge to the right. As in the solution above, the interesting rays are the tangent rays from lights to circles. The interesting moments are when the x-coordinate of the vertical line reaches one of the following. (1) A light. (2) A pillar starts or ends. (3) An interesting ray touches or intersects a circle, or hits the wall. (4) Two interesting rays intersect.

Now, let $x_1 < x_2$ be two adjacent interesting moments. The vertical strip between x_1 and x_2 is divided into pieces. Each piece is bounded above and below by a general segment -- a line segment or an arc. By the definition of the interesting moments, nothing interesting will happen in the middle, and each piece is of one color. So we can sample an arbitrary point from each piece to decide the color. The pieces are not convex, but this is not a problem -- they are convex on any vertical line so we can easily find a point that is inside each piece. The area of each piece is also easy to compute -- it is basically a trapezoid, possibly degenerating into a triangle if

the upper and lower boundaries meet at one end, and one needs to subtract a disk segment for each arc-boundary.

Line sweeping is often used with nice data structures to achieve good complexity. But that is not our primary concern here. We used it for the simplicity of the implementation -- the only geometric operations needed here are intersections between lines and circles.

Approximations

The problem requires a relative or absolute error of at most 10^{-5}, while the total room area is 10000. Cases requiring the most care are those when one of the four colors has an area less than 1, in which case the error we can make relative to the area of the whole room is 10^{-9}.

The simplest approach would be to sample a lot of points either randomly, or in a regular grid, compute the color of each sample and assume that the sample is representative of the correct answer. The above error estimation suggests though that to get enough precision, we would need to sample on the order of 10^9 points (or more, due to random deviations). This is too much for a solution to run within the time limit. A smarter approach is needed.

Computing the area can be seen as a problem of computing a two-dimensional integral. A hybrid approach is also possible: we can see it as a one-dimensional integral along the **x** coordinate, and for each **x** coordinate we sample we can compute the exact answer by looking at which segment of the vertical line is in what color. This one-dimensional sub-problem is somewhat simpler to do than solving the full two-dimenstional problem exactly.

In either case, whether we compute a two-dimensional integral or just a one-dimensional one for a more complex function, we need a smart way to approximate the integral. Uniform or random sampling is not enough.

You can search the web for methods of numerical integration. In this problem, an adaptive algorithm is needed, which means that we sample more "interesting" areas with more samples than the less "interesting" ones. "Interesting" can be defined as "large changes in values" (large first derivative) or "wild changes in values" (large second derivative).

One simple algorithm is to write the integration procedure as a recursive function. We recursively try splitting the interval into smaller ones, and see how much the answer changes through such increases of precision. We stop the recursion when the answer changes very little, which means the interval is small enough or the function is smooth enough in the interval. This will result in sampling the more "interesting" areas more accurately.