Problem Title Description. Problem Name: Entmoot Author's Name: Enclosing Circles Problem Code: BALL Alphabet: C

Problem: You have N points in the plane (x_i, y_i), and with each you associate a speed s_i. The question then asks, what is the earliest point in time t in which all these points can travel to a common "meeting" point. Output this t.

Solution: There are two approaches that can be used.

Approach 1: Convexity (of functions).

In this approach, we need to understand the concept of convex functions. A convex function is one in which any line segment joining two points on the function, lies above (not necessarily strictly) the value of the function. Mathematically, this is described as: $f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$ $\forall x, y$ and $t \in [0, 1]$

Some properties of convex functions are: 1) Local optima <=> global optima. 2) Second derivative is always "positive"

Given a convex function f, if I were to ask you for the minimum value it achieves (or where it achieves it), a standard approach used is *ternary search*. Ternary search basically looks something like this:

```
Initialize low, high;
while(high - low > EPS)
    mid1 = (2*low + high)/3;
    mid2 = (low + 2*high)/3;
    if(f(mid1) >= f(mid2))
        low = mid1;
    else
        high = mid2;
return low; //or high, or (low+high)/2: basically they're all within EPS
```

While the above discussion seems to be for *functions of one variable*, the same holds for even *functions of multiple variables*, where you let x and y be 2-dimensional points (or point-vectors that allow for addition and scaling). Or in fact for any higher dimension as well: just the co-domain is in Reals.

In multidimensional space, (say 2 dimensions), ternary search would actually look like *nested* ternary search. This is because, for fixed x, f will be convex in the one-dimensional y. And further, optimum values you get from x, will also be convex in x.

Nested ternary search would work as follows:

```
//Let usmakea functiong(x) which returns optimum value f(x, y)
                when varied over y
```

```
g(x) :
Initialize lowy, highy;
while(highy - lowy > EPS)
      mid1 = (2*lowy + highy)/3;
      mid2 = (lowy + 2*highy)/3;
      if(f(x, mid1) >= f(x, mid2))
          lowy = mid1;
      else
          highy = mid2;
return f(x, lowy);

optimum():
Initialize lowx, highx;
while(highx - lowx > EPS)
      mid1 = (2*lowx + highx)/3;
      mid2 = (lowx + 2*highx)/3;
      if(g(mid1) ¿= g(mid2))
          lowx = mid1;
      else
          highy = mid2;
return g(lowx); //or lowx etc, as you see fit
```

With this background of ternary search and convex functions (which do turn up a lot in the course of computational geometry), let us go ahead applying this theory and solving this problem.

Firstly, let $f_i(x, y) =$ square of time taken for the ith point to reach (x, y). This is in fact, just

$$f_i(x, y) = \frac{(x - x_i)^2 + (y - y_i)^2}{s_i^2}$$

Now, to actually get the (square of) time at which the meeting will begin *at this point*, we just take the max of these f_i values. Let

$$f(x, y) = \max_i f_i(x, y)$$

Minimizing f, i.e. minimizing the square of the time taken is equivalent to minimizing the time taken itself.

Finally, it is easy to *see* that all the f_i are convex. Is f also convex? Indeed it is!

Proof: Let x and y be two arbitrary points in the plane
$tf(x) + (1-t)f(y)$
$\geq tf_i(x) + (1-t)f_j(y)\ \forall i, j$ and $t \in [0,1]$
$\geq tf_i(x) + (1-t)f_i(y)$ (above holds for $i = j$ in particular)
$\geq f_i(tx + (1-t)y)$

Which shows that

$$tf(x) + (1-t)f(y) \geq f_i(tx + (1-t)y)\ \forall i$$

and hence

$$tf(x) + (1-t)f(y) \geq \max_i f_i(tx + (1-t)y) = f(tx + (1-t)y)$$

and hence, f is also convex. Using nested ternary search on f now, we will be able to find the minimum possible time as required.

Approach 2: Event points.

The technique used here is a very widely used technique in solving geometry problems. It helps take a continuous search space and converts into a a discrete space, usually through a series of "without loss of generality" arguments.

Here, let us as a function of time t, draw circles centered at each given point (x_i, y_i) of radius $t * s_i$. Now, it is possible for every point to meet in time t iff these circles' intersection is non-empty. Indeed, any point that belongs to any particular circle can be reached in time t from that circle's center; and hence, an intersection point corresponds to all points being able to meet.

Now, "without loss of generality", the optimal meeting point does not lie in the interior of all the circles. If it did, then it would have been possible shrink all of the circles by reducing the time t by a small amount δt. Hence, the supposed point would still be in the intersection of all the circles, but now it can be reached in a shorter time!

Further, "without loss of generality", the optimal meeting point does not lie on the boundary of only one circle. If it did, again by the shrinking process, here we would now also move our "optimal point" inwards in this circle, but we would still have it in the intersection of all circles even after decreasing t by δt

Now, we have that the optimal point lies on the boundary of atleast two circles. Lets see where our shrinking argument takes us if it is exactly two circles. Consider two circles, and lets say they intersect at two distinct points. Now, by shrinking it further by a small amount, and moving your supposed point along the boundary of the circles' intersection, you would still have it as being in the intersection of all the circles, since for all other circles our

initial supposed point was actually in the interior. Hence this case does not give us our optimal point.

What if the two circles' boundaries just touch? Then further shrinking would cause these two circles to not intersect any more. Hence this is a possible optimal point.

The other case is for optimal points on atleast three circles' boundaries.

Now, how do we *find* these candidate points? For the two circles' case it is easy. Given (center) points O_1 and O_2 with speeds s_1 and s_2, the candidate point is that unique point P on the line segment O_1O_2 dividing it in the ratio $s_1 : s_2$.

Now, how do we find the intersection of three circles? Let us ask ourselves, what is the locus of the point P such that it lies on the boundary of circles centered at O_1 and O_2 for two such points. i.e. We want to find the locus of the point P such that $PO_1 : PO_2 = s_1 : s_2$.

For $s_1 = s_2$, the locus is merely the perpendicular bisector of O_1O_2. With a little bit of coordinate geometry (or if you remember your high school geometry), then you will get that for $s_1 \neq s_2$, the locus of P is actually a circle itself. And we know two diametrically opposite points of this circle : they are the two points P on the line joining O_1O_2 satisfying the given speeds' ratio.

Finally, we need to put together all these circle-circle / line-line / circle-line intersection algorithms to give us a list of candidate/event points for our answer. Once we have these points, and the corresponding times, then we can just calculate which among them is feasible and at the minimum time point, which gives us our answer.

Time Complexity: $O(N^3)$ for finding candidate points, and $O(N)$ for evaluating times for each point, giving overall complexity of $O(N^4)$.