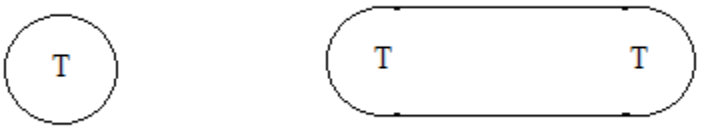
COP 4516 Spring 2022 Week 12 Team Contest #4 Solution Sketches

Euclid

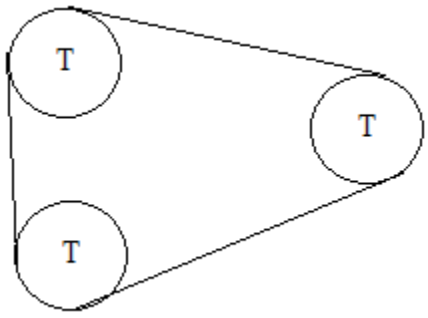
Take in the input and calculate the necessary vectors: AB, AC, DE and DF. From these vectors, use the cross product magnitude to get the areas of triangle DEF and the parallelogram defined by A, B and C. Divide these two areas to get a "scale factor" between the two figures. If ABC is "big" then this scale factor will be less than one. Multiply the scale factor by the vector AC. (To multiply a vector by a scalar, just multiply each component of the vector by that scalar.) This will be the vector AH. Since point A is known, just add vector AH to point A to obtain point H. From there, add vector AB (which is equal to vector HG) to point H to obtain point G.

Farmer John's Forest

It should be fairly clear that a convex hull is required to enclose all of the "outer most" trees with minimal fencing. However, the "extra margin of error seems quite problematic. One thing to notice is that in order for the tree to be exactly a distance of c away from the fence, at least a portion of the fence will be in the shape of a circle. Consider the case for one or two trees, drawn below:



In the first case, the fence is exactly a circle with radius c. in the second case, it appears as if both curved parts are semicircles with radius c. Now let's look at three trees but where I leave the pictures of the circles around the trees in.



If we carefully look at this picture, it appears as if each curved part is a "different" section of the circle and if we look at all the curved parts on the outside, together they form one circle. If you use some standard geometry, (sum of the external angles of a convex polygon equals 2π radians), then you see that this observation is true. Namely, the sum of the lengths of all the curved parts of the fence will just be a single circle of radius c. Thus the solution is as follows: (1) find the convex hull of the points, (2) determine the sum of the lengths of the edges on the convex hull, (3) add to the value from (2) the circumference of a circle with radius c, where c is given in the input.

Presidential Security

The least cost connection between a set of nodes is simply the minimum spanning tree of the induced graph. Each room is a vertex. The edge weight between rooms is simply sum of the absolute values of the difference of floors and the absolute values of the difference of rooms. We can extract the floor by doing /100 and the room by doing %100.

Interesting Intersections

The solution to this problem was given in Tuesday's lecture. You just have to be careful with corner cases. The circle's equation is of the form $(x-a)^2+(y-b)^2=r^2$. You can express the equation vector equation of the line and get something of the form $x=c+d\lambda,\ y=e+f\lambda$. In these equations, *a*, *b*, *c*, *d*, *e* and *f* are known based on the input. Substitute the latter expressions for x and y into the circle to get:

$$(c+d\lambda-a)^2+(e+f\lambda-b)^2=r^2$$

This is a quadratic in λ. Solve for λ and see if either root is in between 0 and 1 (on segment). It's possible that there is no real solution for λ, which simply means that the line itself doesn't intersect with the circle. You have to check for this to avoid mathematical errors by checking the discriminant of the quadratic before you take the square root of it.

Squirrel Territory

If the radii of two circles are equal to r units, then the circles intersect if and only if their centers are less than or equal to 2r units apart. So, the closer two centers are, the further restricted the radius is. Thus, we aim to find the closest pair of centers (tree locations) and divide this value by 2. Since the data is small, a double for loop through all pairs of tree locations is sufficient.

Fujiyama Thursday

This is probably the easiest problem in the set. Since the cars all leave at the same time, we want the students who eat the fastest to go in the slowest car and the students to eat the slowest to go in the fastest car. So, sort the driving times of the cars and the eating times of the students (two separate sorts). For implementation, reverse the array storing the student eating times, so the slowest student is in index 0. From there, just place the first four students (slowest) in the fastest car, keeping note of when the last student will finish eating.