Flow Contest Editorial

Maxim Akhmedov

September 29, 2018

1 Problem A. Elementary

Just implement the mixed product formula. The volume of the tetrahedron is 1/6 of the mixed product of its three side vectors.

2 Problem B. Cross Spider

You need to check if all points belong to the same plane. If $n \leq 3$, the answer is positive. Otherwise, pick first two points and check if there is at least one point that is not collinear with them using the vector product. If so, find an equation of the plane passing through these three points (using the vector product) and check if all of the remaining points belong to it.

3 Problem C. Line Teleportation

Build a graph consisting of start point, end point and each of the teleportation lines, calculate the pairwise distances between them and run a Floyd-Warshall algorithm (or any other shortest path algorithm). Do not forget about the possibility of getting from the start point directly to the end. As a primitive, you are required to calculate the distance between a point and a line and a distance between two lines which were discussed during the lecture.

4 Problem D. Segment Distance in 3D

There are two approaches for this problem: use two ternary searches or modify an algorithm for finding the shortest distance between lines. In case when segments are not parallel, find the shortest segment connecting these two lines and then move its endpoints towards the corresponding segments (if needed). In case when they are parallel, check if two segments “overlap“ when one line is projected onto another; if yes, the answer is the same as for lines, otherwise the answer is the shortest distance between the endpoints of the segments.

5 Problem E. 3D Printing

The main difficulty in this problem is to read its statement carefully. There is an important constraint in the input format section: “None of the polyhedra in any given test case will overlap". After reading that it should become clear, that all you need is to implement the polytope volume calculation procedure which is pretty simple in case of this problem since the polyhedra are defined by their faces. Still, you need to find a correct orientation for each of the faces before summing up signed pyramid volumes.

6 Problem F. 3D Model

This problem requires you to build a convex hull of given points and then for each face find how many vertices does it have. Note that it is different from the number of points that are located inside the face because the point inside the face cannot be considered to be its vertex. So, you may start with building the convex hull using the naive approach in $\mathcal{O}(n^4)$, and then for each of the faces you have to build a 2-dimensional convex hull of points inside it.

In order to build a 2-dimensional convex hull, you may use one of two approaches. First one is transforming 3d coordinates of all points inside face into 2d coordinates by choosing three non-collinear points $\vec{p_1}, \vec{p_2}, \vec{p_3}$ inside the face, defining basis vectors $\vec{f_x} = \vec{p_2} - \vec{p_1}$ and $\vec{f_y} = \vec{p_3} - \vec{p_1}$ and then calculating coordinates of a point $\vec{p_i}$ by formulae $x'_i = \vec{p_i} \cdot \vec{f_x}$, $y'_i = \vec{p_i} \cdot \vec{f_y}$. After doing so, run a regular 2d convex hull algorithm. Note that with this approach 2d coordinates will have magnitude of C^2 where C is the magnitude of the coordinates of the original points, so when you build a 2d convex hull you will be dealing with vector products of magnitude C^4. It is ok for this problem, but it may be done more efficiently.

The more optimal approach is to run 2d convex hull algorithm on the original 3d points without moving to 2d space. All we need is the predicate "vector $\vec{v}$ is counter-clockwise direction in respect to the vector $\vec{w}$". Suppose that $\vec{p_1}$ is some point inside the face and $\vec{p_0}$ is some point outside the face. Let $\vec{q} = \vec{p_0} - \vec{p_1}$. Now it is easy to see that we can use $\mathrm{sgn}[\vec{v}, \vec{w}, \vec{q}]$ as a predicate for convex hull algorithm and this predicate requires only calculations of magniutde C^3 and does not requires finding three non-collinear points inside the face, calculating 2d coordinates and implementing structures for 2d geometry.

As a final remark, it is even possible to reduce the magnitude of calculations down to C^2 if we notice that among basis vectors $\vec{e_x}, \vec{e_y}, \vec{e_z}$ (vectors with zero at all coordinates except the one specified in the index where there is 1) at least one does not belong to our face and it can be taken as $\vec{v}$ in the last approach.

7 Problem G. Asteroids

Start with finding the faces of each of the polytopes. Calculate the center of masses for each of the polytopes by dividing each of the polytopes into tetrahedrons. Finally, calculate the shortest distance from the center of masses to the closest face to it. Note that it is equivalent to simply finding the perpendicular length onto all planes of the faces and you do not need to check if perpendicular belongs to the face. Indeed, if it does not belong to face, it means that it got out of the polytope somewhere else and the chosen face is definitely not the closest one. Now it is obvious that the answer is the sum of these two shortest distances for our two polygons.

8 Problem H. Connected Rings

Consider two planes containing the rings. If these planes are parallel, the answer is negative. Otherwise there is a line l in the intersection of these two planes. If rings are connected, then each of the circles should intersect l in two points and these four points should form an incorrect bracket sequence (for example, a_1, b_1, a_2, b_2 where a_i are the intersection points of the first circle and b_i are the intersection points of the second circle with a line).

Now all we need is to find these intersection points. Intersection points of a first circle with l may also be defined as intersection points of the first circle with the plane of the second circle. The may be found by careful investigation of several right triangles that I am too lazy to describe here but will definitely show on the board.

9 Problem I. Black Hole

This is almost not a 3d geometry problem, but still a rather beautiful problem. It is easy to see that z-coordinates do not have any meaning in this problem, so we move to Oxy plane and we are interested if it is possible to contract given polyline into a distant point on a plane with n points removed.

The first idea that comes to our head is to calculate the index of each point in respect to a given polyline, i.e. the number of rotations the polyline make around each of the points. If the answer is positive, it should be zero for each point (otherwise we will stuck around this point). Unfortunately, the contrary is incorrect: consider four points $A = (0, 0)$, $B = (4, 0)$, $C = (4, 4)$, $D = (0, 4)$, the two cut points $p_1 = (1, 2)$ and $p_2 = (3, 2)$ and the closed polyline $ABCADCBD$. It is easy to verify that both p_1 and p_2 have index zero in respect to this polyline, i.e. if there was only the cut point p_1 or only the cut point p_2, it would be easy to contract the polyline into the point. But when there are two cut points, and a given polyline, it is impossible.

Underlying mathematical object describing this fact is called a *fundamental group* of a plane with n cut points. In case of plane without n cut points the fundamental group is a free group with n generators that is not Abelian, which was the mistake of the previous solution.

If we stop using the scary mathematical words, the idea is that we need to know the story of how the polyline rotates around all the points without losing the order of these events. Namely, put a ray out of each of the points such that these rays do not intersect (for example, a ray with a direction vector $(1, 10^5)$). Now when crossing this ray for a point p_i from one side (for example, from below), write down a term p_i, and when crossing it from the other side, write down a term p_i^{-1}. We got a string consisting of characters p_i and p_i^{-1}. We are allowed to transform it by inserting or erasing two adjacent characters of form $p_i p_i^{-1}$ or $p_i^{-1} p_i$ and the question is can we reduce this word to an empty word. It may be proven that we just need to discard all pairs of characters of described form until it possible, and if it is not possible any more, we check if we got an empty string.

10 Problem J. The Worm in the Apple

In this problem you are required to build a convex hull in $O(n^2)$ using the gift wrapping method and then find the closest face for each of the query points in a linear time.

11 Problem K. Spheres

Consider the sought plane π. It has a property that $dist(\pi, O_i) = \pm r_i$. Fix one of 8 ways to put signs behind r_i. Now we got a system of three linear equations over four variables A, B, C, D defining the plane π with an extra equation of $A^2+B^2+C^2 = 1$. Discarding the last equation, we get three linear equations over four variables and it may be shown that these linear equations are linearly independent because centers of the spheres are non-collinear. Solve this system, the solutions form an 1-dimensional subspace that is a line. Find any non-zero point on this line and then normalize it by dividing by $\sqrt{A^2 + B^2 + C^2}$.

12 Problem L. Cubes

Consider each of 12 planes of the faces of two cubes. Let's find out how the face of the resulting polytope looks like on this plane. It is defined by an intersection of half-spaces inside that plane (each of which is an intersection of a half-space defined by some of the original faces and the fixed face plane).

13 Problem M. Another Short Problem

Recall the Euler Formula: $V - E + F = 2$. It is correct only if there are least three non-collinear points in our convex hull.

Note that in our case $3F = 2E$ because each face of the convex hull is triangular and each edge is adjacent to exactly two faces. Thus, we have an equation: $V = F/2 + 2$.

Let us tune this equation a bit to make it work even for cases when there are 2 points or less. If there are 2 points, $V = 2$, $F = 0$ and the formula is correct by a strange coincidence. If there is 1 point, $V = 1$, $F = 0$ and the right side should be decreased by 1, and if there is 0 points, the right side should be decreased by 2. So, we can write a tuned Euler Formula: $V = F/2 + 2 - T$ where T is equal to 1 if there is only 1 point and to 2 if there are no points at all.

Now let us apply this formula to the problem. We want to calculate the expected value of V. Due to linearity of expected value, it is equivalent to calculate the expected values of F and T. Expected value of T may be easily calculated by definition. Expected value of F is a sum over all oriented triples of points p_i, p_j, p_k of a probability that $p_ip_jp_k$ is a (counter-clockwise oriented) face of the resulting convex hull which is equivalent to the fact that behind the plane $p_ip_jp_k$ all points were not included and p_i, p_j, p_k were included.

Thus we get a solution with a running time of $O(n^4)$.