Opracowanie: ARS
Ars Longa

HISTORIA:
• v. 1.0: 2005.xx.xx, [SA] xxx

dokument systemu SINOL 1.5

1 Introduction

During ACM world finals, sometimes there are physics-related problems. Usually, they do not need any deep physical knowledge. Usually the knowledge gained in secondary school is enough. This is the case here too.

Usually physics problems were solved by calculation one or more physical quantities. This task, at the first glance, looks a bit differently — they want us to just determine whether some physical system is static or even stable. How can we start solving such a problem?

Let's look at the forces present in this system. For consistency, let's denote the consecutive balls $K_1, K_2, \ldots, K_n$. Here are the three types of forces, which are interesting for us:

- forces of gravity, acting on all the balls

Rysunek 1: Gravity acts on every ball in the system.

- forces of elasticity; sometimes called also forces *carried by rods*

Rysunek 2: A rod being stretched is acting on the balls according to the 3'rd Newton's law.

  Directions of there forces are the same as directions of the rods. Each rod is acting on the balls attached to it using forces of equal values, but opposite directions. Let's denote by f_i the value (in newtons) of a force carried by i-th rod. Obviously, we treat the length of a vector as being the value of the force represented by the vector.

- table reaction forces, acting only on the balls directly on the table, which equillibrtes all the other forces acting on these balls

Rysunek 3: Table reaction force. Please note that in this example, the rods number 2 and 4 are compressed, so we have $f_2 f_2 < 0$.

2 The first solution

So? Are we done now after we know what are the forces? Isn't it enough to calculate all the forces, check, whether all the forces acting on each ball sum up to zero to know wheter the construction is static or not? (Let's leave the second problem of checking if a static construction is stable or not for a later time.)

More or less, this is an inspiring idea, but it needs some refinements. At first, let's note that sometimes the forces carried by rods are not unique and one can determine multiple systems of such forces and all of them will describe the system equally well. This sounds strange, as th classical physics we never had such uncertainty problems. But here we have an assumption, that rods are not compressible. So look at the figure below and try to think about forces, which act there.

Rysunek 4: An example of a construction, where forces carried by rods are not determined uniquely.

The second problem is that calculating the forces in general is not a simple task. But fortunately, this is not needed as well. Let's go this way: *let's assume that the system is static* and let's try to calculate the forces carried by all rods. The good point about it is that we can desctibe this task in a clean mathemathical way — using a system os equations.

But before doing this, let's add some more denotations. Let $R(K_i)$ be a set of numbers of all the rods connected to K_i. Moreover, let $b(i, K_x)$ be a numer of a ball, which is connected to the i-th rod, but which is not K_x.

Now, for each ball, let's write down the fact, that the sum of all the forces acting on this ball is zero:

$$\begin{cases} \sum_{i \in R(K_1)} f_i \cdot \dfrac{\overrightarrow{K_1 K_{B(i,K_1)}}}{\left|\overrightarrow{K_1 K_{B(i,K_1)}}\right|} = -\vec{g} \\ \sum_{i \in R(K_2)} f_i \cdot \dfrac{\overrightarrow{K_2 K_{B(i,K_2)}}}{\left|\overrightarrow{K_2 K_{B(i,K_2)}}\right|} = -\vec{g} \\ \vdots \\ \sum_{i \in R(K_n)} f_i \cdot \dfrac{\overrightarrow{K_n K_{B(i,K_n)}}}{\left|\overrightarrow{K_n K_{B(i,K_n)}}\right|} = -\vec{g} \end{cases}$$

Please note that this is a system of vector equations, therefore is we assume everything is in a three-dimensional space, then we can replace each vector equation with three scalar equations (by coordinates). Treating all f_i as unknowns, we get a system of $3 \cdot n$ linear equations with m unknowns.

Now we use a well-known Gauss elimination algorithm and we convert the matrix of this system to a canonical triangular form. If at that point we know, that the system has no solutions, then we know that our assumption about staticity is false (because is it impossible to determine valis forces carried by rods). So at this point we know how to determine whether the system is static or not. We are left with the secnd problem — determining whether a static system is stable or not.

Let's formulate this problem in terms of our equations. We ask wether exist such small vectors

$\vec{\alpha_1}, \ldots, \vec{\alpha_n}$, that the system

$$\begin{cases} \sum_{i \in R(K_1)} f_i \cdot \frac{\overrightarrow{K_1 K_{B(i,K_1)}}}{\left|\overrightarrow{K_1 K_{B(i,K_1)}}\right|} = -\vec{g} + \vec{\alpha_1} \\ \sum_{i \in R(K_2)} f_i \cdot \frac{\overrightarrow{K_2 K_{B(i,K_2)}}}{\left|\overrightarrow{K_2 K_{B(i,K_2)}}\right|} = -\vec{g} + \vec{\alpha_2} \\ \vdots \\ \sum_{i \in R(K_n)} f_i \cdot \frac{\overrightarrow{K_n K_{B(i,K_n)}}}{\left|\overrightarrow{K_n K_{B(i,K_n)}}\right|} = -\vec{g} + \vec{\alpha_3} \end{cases}$$

is no more solvable. The problem is that we do not know exactly, what are the values of α_i. Here are two ideas, which can help us:

1. One can try to generate randomly small α_i vectors and then use the Gaussian elimination once again to confirm whether the system is still solvable or not[1]. If it is not, then we can be sure, that the construction is unstable. If the system remain solvable, then one can try again with some other vectors or give up, answering that the construction is static. Probability of obtaining a wrong answer is very small with this algorithm.

2. Maybe it is not needed to use such an ugly non-deterministic hack. Let's look carefully on what we have obtained in the first phase. This is the canonical triangular form of the extended matrix. It looks like this:

$$\left|\begin{array}{ccc|c} a_{1,1} & \cdots & a_{1,m} & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & a_{k,m} & b_k \\ 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{array}\right|$$

   At the beginning we kave k non-sero rows and then we *may* have some zero rows. If now we add to the original system of equations the α_is, then the canonical triangular matrix form will look like this:

$$\left|\begin{array}{ccc|c} a_{1,1} & \cdots & a_{1,m} & b_1 + \Delta_1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & a_{k,m} & b_k + \Delta_k \\ 0 & \cdots & 0 & \Delta_{k+1} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & \Delta_{3n} \end{array}\right|$$

   where, beware, Δ_i are nonzero linear combinations of coordinates of α_i vectors. So when exist such α_is, that the system is no more solvable? Here's the answer: if all the rows are non-zero, then the system is always solvable. Otherwise it is also possible to find such α_is, that let's say Δ_{k+1} is non-zero. And this implies unsolvability of the system.

[1] Actually, it does not matter whether the vectors are "small" or not. Having a system of destabilizing forces, we can multiply each of them with an arbitrary constant, and we will still have a destabilizing system of forces.

3 Additional refinements

Algorithm described above is a correct solution of this problem. But in an atmosphere of a contest, just finding a correct, and even efficient, solutions is not always enough. The challenge is to implement it in such a way that we are sure that if our implementation is correct, then our solution will be accepted.

I would be reluctant to implement the base algorithm presented above directly. I wouldn't be sure enough, that the algorithm is numerically stable. Moreover I may not have the more numerically stable version of Gauss elimination known as .... If I would get `Wrong answer` I would never be sure whether the problem lies in some bug floating in the code or because of the numerical issues.

Therefore I'm going to create such a solution, in which I never use floating-point values. At first, we can assume that the values in the input do not have more than 20 decimal positions. This lets us store them as integers. So now coordinates of all vectors in the equations have integer coordinates. Unfortunately, the values in denominators in the equations are not only not integers, but may be even irrational. But we have a solution for this too. An equation

$$\sum_{i \in R(K_1)} f_i \cdot \frac{\overrightarrow{K_1 K_{B(i,K_1)}}}{\left|\overrightarrow{K_1 K_{B(i,K_1)}}\right|} = -\vec{g}$$

can be rewritten as

$$\sum_{i \in R(K_1)} g_i \cdot \overrightarrow{K_1 K_{B(i,K_1)}} = -\vec{g}$$

where now $g_i = \frac{f_i}{\left|K_1 K_{B(i,K_1)}\right|}$ is treated as unknown.

Unfortunately, this is not all yet. Operations on matrices cannot be done in integers, as integers do not form a mathematical field (division is not always defined). But we can eventually use some other field, for example $\mathbf{Z}_p$ for some large prime p, e.g. $2^{31} - 1$. This is some nasty hack though, as solvability of a system in $\mathbf{Z}_p$ does not imply its solvability in $\mathbf{Q}$. But for us this works well enough. It is a good exercise to calculate some ... probability, that a random matrix in $\mathbf{Z}_p$ is invertible, but it is not invertible in $\mathbf{Q}$. Even harder it is to find such a counterexample, which works at the same time for multiple different values of p.

The presented algorithm, together with all refinements, was implemented and can be found among the resources provided together with this document.

4 Tests

TODO