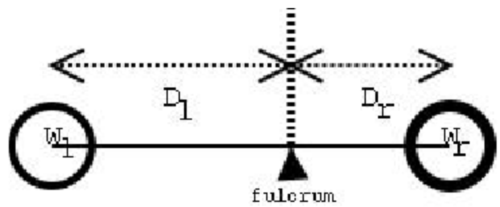
Tianping Not So Mobile UVA 839

tags: Simple algorithm (/tag/Simple+algorithm/)

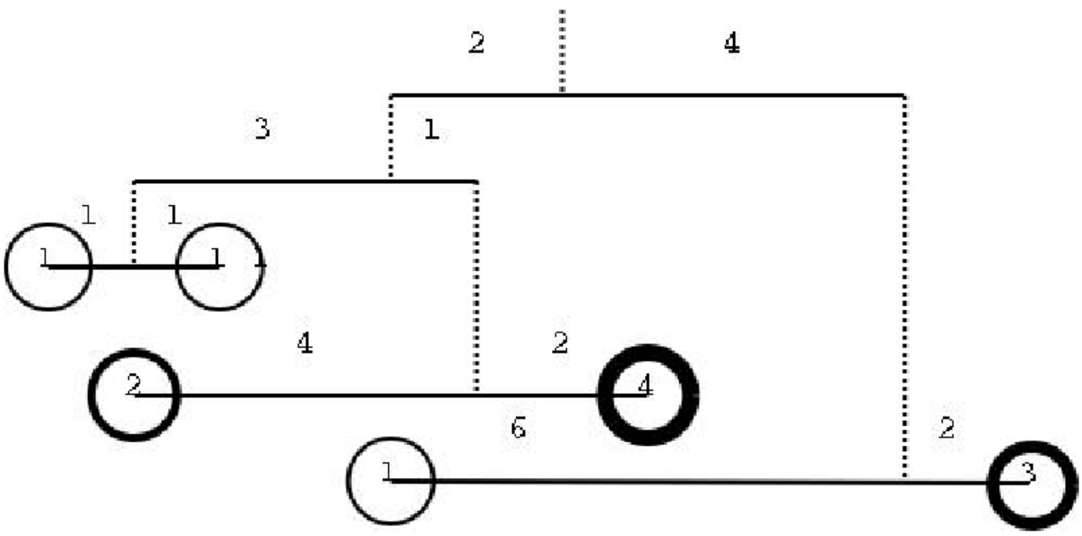
Topic description

Before being an ubiquous communications gadget, a *mobile* was just a structure made of strings and wires suspending colourfull things. This kind of mobile is usually found hanging over cradles of small babies.



The figure illustrates a simple mobile. It is just a wire, suspended by a string, with an object on each side. It can also be seen as a kind of lever with the fulcrum on the point where the string ties the wire. From the lever principle we know that to balance a simple mobile the product of the weight of the objects by their distance to the fulcrum must be equal. That is $W_l \times D_l = W_r \times D_r$ where D_l is the left distance, D_r is the right distance, W_l is the left weight and W_r is the right weight.

In a more complex mobile the object may be replaced by a sub-mobile, as shown in the next figure. In this case it is not so straightforward to check if the mobile is balanced so we need you to write a program that, given a description of a mobile as input, checks whether the mobile is in equilibrium or not.



Input format

The input begins with a single positive integer on a line by itself indicating the number of the cases following, each of them as described below. This line is followed by a blank line, and there is also a blank line between two consecutive inputs.

The input is composed of several lines, each containing 4 integers separated by a single space. The 4 integers represent the distances of each object to the fulcrum and their weights, in the format: W_l D_l W_r D_r

If W_l or W_r is zero then there is a sub-mobile hanging from that end and the following lines define the the sub-mobile. In this case we compute the weight of the sub-mobile as the sum of weights of all its objects, disregarding the weight of the wires and strings. If both W_l and W_r are zero then the following lines define two sub-mobiles: first the left then the right one.

Output format

For each test case, the output must follow the description below. The outputs of two consecutive cases will be separated by a blank line.

Write 'YES' if the mobile is in equilibrium, write 'NO' otherwise.

Test

Before becoming an AC tool, the wind chime is only a structure of a colorful item. It usually hangs on the baby's cradle. The following explanation shows a simple wind chime. It is just a lever, hanging on the rope, and there is an object in both sides. It can also be seen as a balance. From the principle of leverage, we can know that the balance condition of a lever is WL × DL = Wr × DR. (Where WL is the weight of the left object, the length of the left arm is the length of the left arm, the weight of the right object, the length of the DR is the length of the right arm) A more complex balance can also be calculated in this manner. As shown in Figure 2, in this case, calculating whether the balance balance is very complicated, so we need you to write a program, give a balance as an input, calculate whether the balance is in a balanced state.

Enter the first behavior of the first behavior n, which represents the N group inquiry, and each reference is described below. There is a blank line behind this line, and there is a space between two consecutive inputs. The input consists of several lines, and each line contains four integers separated by a single space. These four integers are represented by WL, DL, WR, DR; WL, DL WR DR, if WL or WR is zero, and there is still subtyfline below WL or WR. In this case, we calculate the weight of the sub-flat as the weight of the object (WL or WR), the weight of the lever and the line. If WL and WR are zero, then give the left sub-flat and give the right day.

Output For each inquiry, two consecutive queries are separated from one blank line. If the balance is in a balanced state, output "YES", otherwise "NO" is output.

[Input sample]

```
1
0 2 0 4
0 3 0 1
1 1 1 1
2 4 4 2
1 6 3 2
```

[Output sample]

```
1
0 2 0 4
0 3 0 1
1 1 1 1
2 4 4 2
1 6 3 2
```