
ACM Programming Challenges Lab

Exercise 3 – Wild West Train Network

You, the fastest computer scientist in the whole wild west, have just invented a revolutionary steam powered computing device. As often the case with inventors working out of their family's barn, you have no money to commercialize your creation and you need an investor to help you out. Several rich gentlemen from distant cities showed some interest, but all would like a demonstration of your machine first. You decide to go and visit them, but given the vast number of private train companies and their myriad of connections, finding the fastest route to them is usually a nightmare... except, that you have just invented this great computing device which can solve this problem within minutes and less than two bucketfuls of coal and water!

You want to just give your device a list of all train networks in the general direction of the city a given investor lives in, and get out the fastest route to each of them. Curiously spiders seem to really enjoy nesting in your machine, which makes it go slow. For this reason, after each demonstration you need to return back to your workshop to remove these nasty bugs from your machine before you go on to the next destination.

Input You decide that the machine should read the train connections off a piece of paper in the following way. On the first line there will be an integer c , the number of cases (i.e. rich gentlemen). Then c cases will follow, each one starts with a line with four numbers n, m, y and i , where $2 \leq n \leq 20000$ and $0 \leq m \leq 50000$. This means that there will be n cities in the train network and that m train tracks connect them. The numbers $0 \leq y < n$ and $0 \leq i < n$, denote your and the investor's city respectively. Clearly $y \neq i$. The next m lines contain 3 numbers each, two numbers of cities connected by a direct train route and the time it takes to travel between them (which is the same in both directions).

Output For each investor print the fastest possible time it takes to get there in the given train network. If you find that there is no possible connection then print "Need a horse" (note the capitalization). Put the output for each case on a separate line.

Sample Input

```
3
2 1 1 0
1 0 12
3 3 0 2
1 2 12
0 2 100
0 1 42
2 0 0 1
```

Sample Output

```
12
54
Need a horse
```