Challenge 14 - Train Empire

One of your friends (or maybe it was you?) is quite a nerd, and loves trains. He has recently found the game of his life, but after spending some time taking care of his beloved trains, he wants to automate the way they move around the world.

The rules of the game are simple, but there are quite a few and they must be explained. There game has four main elements:

- **stations**: they are located all over the world in certain **coordinates**. There is one wagon waiting at each station.
- **wagons**: they have a certain **value** and are meant to be delivered to a **destination** station.
- **routes**: they consist of a group of connected stations that one train can travel through. A station can be in more than one route. A route can branch out instead of being simply a series of stations. Trains

follow a straight line when going from one station to the next.
- **trains**: they work one route, can only carry one wagon at a time, and have a limited amount of **fuel**.

This is how the score works in the game:

- When a train delivers a wagon to the destination station, it leaves the wagon there, adds its value to the **score** and continues on as long as it has fuel left.
- Wagons can be left at any station, not just at their destination. There is no limit to the number of wagons parked at a station (your friend has purchased all the station expansions!).

Your work is to maximize the score before all the trains run out of fuel.

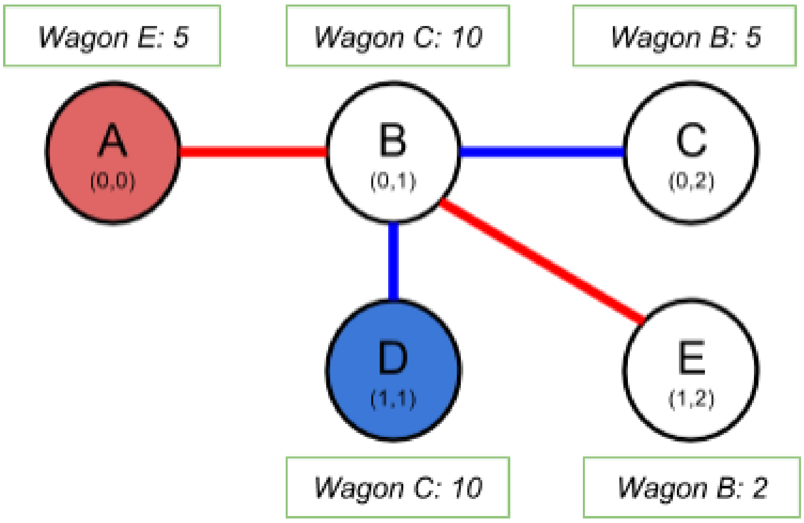
Note: all the numeric parameters are integers, but due to the distance between stations the fuel for each train can decrease in fractional values.

Example

We have 5 stations in two routes: red and blue.

- The red route comprises stations A, B and E. The train starts in A.
- The blue route comprises stations B, C and D. The train starts in D.

Each train has a fuel capacity of 3, and the stations are arranged as follows:



- The train in the red route only has fuel to deliver the cargo in A to E (value 5).
- The train in the blue route can deliver the wagon in D to C (value 10)

and the wagon in C to B (value 5)

That makes a total score of 20 points.

Input

The first line is **N**. **N** scenarios will follow.

Each scenario starts with a line: **S**,**R**,**F**.

- **S** <= 8: number of stations
- **R** <= 2: number of routes (and therefore trains)
- **F** <= 10: the fuel of the trains

Next, **S** lines will follow with the information of each station in the format: **N X**,**Y D V**:

- **N**: the name of the station
- **X,Y**: coordinates of the station in the map
- **D**: wagon destination
- **V**: wagon value

At last, **R** lines will follow with the route for each train **S C C C**...

- **S**: starting station
- **C**: station connection with the format: **S1-S2**. The train can travel in both directions.

All of this will be clearer in the example.

Output

N lines with the maximum score that can be obtained for each scenario.

Sample input

```
1
5,2,3
A 0,0 E 5
B 0,1 C 10
C 0,2 B 5
D 1,1 C 10
E 1,2 B 2
A A-B B-E
```

```
D B-D B-C
```

Sample output

```
20
```

Submit & test your code

To test and submit code we provide a set of tools to help you. Download contest tools if you haven't already done that. You will then be able to test and submit your solution to this challenge with the challenge token.

```
Challenge token: _HHR9xSRUv1OJ4XLRbwS
```

To test your program

```
./test_challenge _HHR9xSRUv1OJ4XLRbwS path/program
```

A nice output will tell you if your program got the right solution or not. You can try as many times as you need.

To submit your program to the challenge

```
./submit_challenge _HHR9xSRUv1OJ4XLRbwS
path/source_pkg.tgz path/program
```

Note that you first need to solve the test phase before submitting the code. During the submit phase, in some problems, we might give your program harder questions, so try to make your program failsafe.

Important: In this phase, you must provide the source code used to solve the challenge and, if necessary, a brief explanation of how you solved it.

Remember **you can only submit once!** Once your solution is submitted you won't be able to amend it to fix issues or make it faster, so please be sure your solution is finished before submitting it.

If you have any doubts, please check the info section.

Go ahead

I'm done! :)

Once you have submitted your code, hit refresh and continue to next challenge.

I'm stuck! :(

Be sure you follow the Tuenti Engineering twitter for updates and possible hints during the contest.

If this challenge is too hard and you are blocked, you will be able to skip it after two hours. Note that **you won't be able to complete it later**, and you have a limited number of challenges to skip.

Finally, if you run out of skips but are still really stuck with one problem, you will be able to skip it after 24 hours.

Challenge status:

Test case	Not done
Solution submitted	Not done
Skip	You still have to wait 0h, 30m and 0s to be able to skip this challenge

Refresh status

Tweet about this! #TuentiChallenge4

Share Follow @Tuentieng