

10810EECS204001

Data Structures

HW4 : Graph

Submit to OJ: #12520

Description

Imagine you are now a student in Tsing Hua University. There are many campuses locate in different places. Students need to take the shuttle bus to commute from one campus to another. Bad news is, not any two campuses have a shuttle bus. Students may need to transfer the bus. Also, if a student wants to change his major to another campus, he can exchange the credits in some proportion. Now, you need to implement some operations to maintain this situation.

Operations:

1. Add : followed by three integers and one floating point $v1$, $v2$, w , r respectively. It means between campus $v1$ and $v2$, there exist a shuttle bus with moving time equals to “ w ” and students can exchange their credits with “ r ” proportion. (Ex: Add 1 2 30 0.9)
2. Delete: followed by two integers $v1$ and $v2$. It means the shuttle bus and exchange credits policy are both canceled between campus $v1$ and $v2$. (Ex: Delete 1 2)
3. Shortest: followed by two integers $v1$, $v2$. It means you want to know the minimum bus moving time from campus $v1$ to $v2$. (Ex: Shortest 1 3)
4. Map: you want to know those shuttle buses that connect every campus and the minimum total moving time. (Ex: Map)
5. ImportantBus: you wonder if there is a shuttle bus which is so important that you cannot go to some campuses without it. (Ex: ImportantBus)
6. CreditExchange: followed by three integers $v1$, $v2$, c . It means you have total “ c ” credits and you want to maximize the final credits from campus $v1$ to $v2$. Notice that when this operation shows, the exchange rate is always between 0 and 1. (Ex: CreditExchange 1 2 200)
7. CreditExchange2: followed by four integers $v1$, $v2$, c , limit. In some rare cases, the exchange rate may more than 1. However, the restriction is that you cannot exchange your credits more than “limit” time and you cannot exchange

from v1 to v2(also v2 to v1) more than once. (Ex: CreditExchange2 1 2 200 4)

Notice

1. CreditExchange and CreditExchange2 will not show in same testcase. More specifically, the algorithm may be different depends on the range of exchange rate.
2. Credits are always integer. In every exchange, you should round down the credits to its floor. (Ex: 19.9 -> 19)
3. You are allowed to use stl except for those direct functions.
4. The total number of campus is no more than 1000.

Input

A testcase will start with an input integer n ($n < 1000$). It means the number of campus.

After first input, there are multiple operations. Some need to print the outcome and some don't.

Output

For operations "Add" and "Delete", you don't need to print anything.

For operations "Shortest" and "Map", you need to print the minimum total moving time. If the output does not exist, print "Error".

For operations "ImportantBus", you need to print "Yes" if there exists one.

Otherwise, print "No".

For operations "CreditExchange" and "CreditExchange2", you need to print the maximum credits you can get under the rules. If the output does not exist, print "0" .

Example

