[QUANT] Manage your Stock Inventory

HackerRank

On a daily basis you are broker facilitating buy and sell of stocks for different hedge funds to support their trading strategies. As a result you are required to manage the stock movements across different accounts.

Primary function is to deliver stocks to the clients based on their needs, any excess stocks that is present can be pledged in "triparty" accounts to be able to raise cash which can be used for client requirements.

Each movement of stock between accounts has a transaction cost associated with it. Lesser the number of movements lesser is the transaction cost. Additionaly, each account belongs to a parent account. Aggregated transaction cost of the movement within the accounts of same parent account is always less than the transaction cost of the movement between accounts of different parent account, and hence should be always prefered. Another way to put this is cost of any number of transactions within the accounts of single parent account is less than cost of single transaction between 2 accounts that belong to different parent entity

Given the daily trading activity, device an alogirthm that will be able to effectively use the stocks to make sure the client demands are met on priority and maximizing the cash value of the asset, with minimum transaction cost possible. (Each row in the output is a transaction/movement.)

Input Format

First line is the number of stocks(n) and the next (n) lines are of format (*stock Id,price*) where *stock Id* is the stock identifier(string) and *price* is the stock price in $USD.

2

P1,2.5

P2,1.25

Post above lines, you are given the number of accounts between which stocks can be transferred(n) and the next (n) lines are of format (*account Id,account type,parent account*) where *account Id* is the account identifier(string), *account type* is the reference data indicating if it is a custody account(CUSTODY) or a triparty account(TRIPARTY) which will give cash in exchange of excess stocks, the third number *parent account* will represent the parent account representing which parent account does the account belong to.

3

Loc1,CUSTODY,1

Loc2,CUSTODY,2

Loc3,TRIPARTY,2

Post above lines, you are given number of lines(n) to read for eligible accounts per stock(indicating in which account can a particular stock be held), followed by (n) lines of format (*stock Id, account Id*), a

stock will have multiple eligible accounts therefore multiple rows for a stock.

6

P1,Loc1

P1,Loc2

P1,Loc3

P2,Loc1

P2,Loc2

P2,Loc3

Post above lines, you are given number of lines(n) to read for eligible flows(directed) per stock(indicating the movement between the accounts that is allowed for a stock), followed by (n) lines of format (*stock Id, source account Id, destination account Id*), which indicates the stock(*stock Id*) can move from *source account Id* to *destination account Id*, a stock will have multiple eligible flows therefore multiple rows for a stock.

7

P1,Loc1,Loc2

P1,Loc1,Loc3

P1,Loc2,Loc1

P1,Loc2,Loc3

P2,Loc1,Loc2

P2,Loc2,Loc3

P2,Loc3,Loc1

Post above lines, you are given number of lines(n) to read for balances of each stock per account followed by (n) lines of format (*stock Id, account Id, quantity*), representing the stock *stock Id* currently has *quantity* number of shares in account *account Id*, if *quantity* positive number, it indicates that you have the stock excess and negative number is the demand of that stock that needs to be fulfilled.

3

P1,Loc1,10

P1,Loc2,-5

P2,Loc1,5

Constraints

all recommended quantity should be positive numbers only

Output Format

Output should be in the format (*stock Id,source acccount Id,destination account Id,quantity*) representing we need to move *quantity* shares of *stock Id* from account *source account Id* to *destination account Id*. The output rows should be sorted in ascending order of *Stock Id*, then *source account Id*, and *destination account Id* at last where each Id is a String.

P1,Loc1,Loc2,5

P1,Loc1,Loc3,5

P2,Loc1,Loc2,5

P2,Loc2,Loc3,5

Sample Input 0

```
2
P1,2.5
P2,1.25
3
Loc1,CUSTODY,1
Loc2,CUSTODY,2
Loc3,TRIPARTY,2
6
P1,Loc1
P1,Loc2
P1,Loc3
P2,Loc1
P2,Loc2
P2,Loc3
7
P1,Loc1,Loc2
P1,Loc1,Loc3
P1,Loc2,Loc1
P1,Loc2,Loc3
P2,Loc1,Loc2
P2,Loc2,Loc3
P2,Loc3,Loc1
3
P1,Loc1,10
P1,Loc2,-5
P2,Loc1,5
```

Sample Output 0

```
P1,Loc1,Loc2,5
P1,Loc1,Loc3,5
P2,Loc1,Loc2,5
P2,Loc2,Loc3,5
```