



*Exercise EX003: Development*

Trains

Version 2.0

You've been asked to write a program to control a network of autonomous mail delivery trains.

Each instance of this problem has:

A network, consisting of a set of nodes, each described by a (string) name.

A set of edges, each of which links two nodes and has an associated journey time in seconds. Edges are undirected and any number of trains can travel along any edge in any combination of orders. An edge is uniquely described by a pair of nodes.

A set of trains in the network, each of which has a maximum total weight it can carry. All trains start off empty and each train has a node where it starts off.

There is a set of packages in the network, each of which has a weight and starts off located at a node, and each of which has a destination node.

The problem is solved when all packages are located at their destination nodes.

We would like you to write (or think about) a program which takes as input:

- A list of node names, [ Node1, Node2, .. ]
- A list of edges, (Name, Node1, Node2, JourneyTimeInMinutes)
- A list of trains (TrainName, CapacityInKg, StartingNode)
- A list of packages, (PackageName, WeightInKg, StartingNode, DestinationNode)

And produces as output a list of moves. Each move should contain:

- The time at which the move occurs, in seconds, W
- The name of a train which moves, T.
- The node that the train starts at, N1
- The names of the packages the train picks up at the start node, P1
- The node that the train finishes at, N2
- The names of the packages the train drops off at the finishing node, P2

For clarity:

P1 and P2 may be empty

N1 may be equal to N2. No time is taken moving a train from a node to itself.

No time is taken to load or drop off packages.

No train may ever carry more than its capacity.

There must be an edge (N1,N2)

The train will take the journey time for that edge to move from N1 and N2 and may not do anything else whilst it is travelling.

Any number of trains may travel down any edge in either direction at once.

Any number of trains may be at any node at any time (nodes have no capacity limit)  
Each train may carry any number of packages, so long as it does not exceed its capacity  
Trains may end the sequence of moves anywhere.  
A train must start its next move at the destination node for its previous move (or at its starting node if it has not yet moved) - trains cannot teleport.

At the end of your sequence of moves, all packages must have been dropped off at their destination node.

We define the solution time as the earliest time at which all packages have been dropped off at their destination node.

Solution S1 is better than solution S2 if the solution time is lower in S1 than in S2.

We would obviously like the best solution possible, but correctness is more important than optimality.

```
// example input
3          // number of stations
A          // station name
B          // station name
C          // station name

2          // number of edges
E1,A,B,30  // route from A to B that takes 30 minutes
E2,B,C,10  // route from B to C that takes 10 minutes

1          // number of deliveries to be performed
K1,5,A,C   // package K1 with weight 5 located currently at station A
that must be delivered to station C

1          // number of trains
Q1,6,B     // train Q1 with capacity 6 located at station B

// example output

// Move Q1 to A via E1, takes 30 minutes.
W=0, T=Q1, N1=B, P1=[], N2=A, P2=[]
// Now move back to B. Takes 30 minutes.
W=30, T=Q1, N1=A, P1=[K1], N2=B, P2=[]
// Move to C and drop off - takes 10 minutes.
W=60, T=Q1, N1=B, P1=[], N2=C, P2=[K1]
// Takes 70 minutes total.
```