原题：

The local commuter railroad services a number of towns in Kiwiland.  Because of monetary concerns, all of the tracks are 'one-way.'  That is, a route from Kaitaia to Invercargill does not imply the existence of a route from Invercargill to Kaitaia.  In fact, even if both of these routes do happen to exist, they are distinct and are not necessarily the same distance!

The purpose of this problem is to help the railroad provide its customers with information about the routes.  In particular, you will compute the distance along a certain route, the number of different routes between two towns, and the shortest route between two towns.

***Input***:  A directed graph where a node represents a town and an edge represents a route between two towns.  The weighting of the edge represents the distance between the two towns.  A given route will never appear more than once, and for a given route, the starting and ending town will not be the same town.

***Output***: For test input 1 through 5, if no such route exists, output 'NO SUCH ROUTE'.  Otherwise, follow the route as given; do not make any extra stops!  For example, the first problem means to start at city A, then travel directly to city B (a distance of 5), then directly to city C (a distance of 4).

1. The distance of the route A-B-C.
2. The distance of the route A-D.
3. The distance of the route A-D-C.
4. The distance of the route A-E-B-C-D.
5. The distance of the route A-E-D.
6. The number of trips starting at C and ending at C with a maximum of 3 stops.  In the sample data below, there are two such trips: C-D-C (2 stops). and C-E-B-C (3 stops).
7. The number of trips starting at A and ending at C with exactly 4 stops.  In the sample data below, there are three such trips: A to C (via B,C,D); A to C (via D,C,D); and A to C (via D,E,B).
8. The length of the shortest route (in terms of distance to travel) from A to C.
9. The length of the shortest route (in terms of distance to travel) from B to B.
10. The number of different routes from C to C with a distance of less than 30.  In the sample data, the trips are: CDC, CEBC, CEBCDC, CDCEBC, CDEBC, CEBCEBC, CEBCEBCEBC.

***Test Input:***

For the test input, the towns are named using the first few letters of the alphabet from A to D.  A route between two towns (A to B) with a distance of 5 is represented as AB5.

Graph: AB5, BC4, CD8, DC8, DE6, AD5, CE2, EB3, AE7

Expected Output:

Output #1: 9

Output #2: 5

Output #3: 13

Output #4: 22

Output #5: NO SUCH ROUTE

Output #6: 2

Output #7: 3

Output #8: 9

Output #9: 9

Output #10: 7

解读：

A

C

B

D

E

5

4

7

3

6

8

8

2

5

图- 1样例图示

通过如下格式提供铁路图（AB5-->A到B单向铁路距离5个单位）：

AB5, BC4, CD8, DC8, DE6, AD5, CE2, EB3, AE7

问题可以分为如下几类：

1. 根据给定路线计算距离（路线不存在则提示：NO SUCH ROUTE）。
2. 给定起点、终点、最大经过站数量给出有多少种路线。
3. 给定起点、终点、刚好经过站数量给出有多少种路线。
4. 给定起点和终点给出最短路线距离。
5. 给定起点、终点、小于最大路线距离给出有多少种路线。

分析：

1. 铁路图数据处理：

实际运用中铁路数据都会存入数据库中，根据计算复杂度、并发量等可以选择oracle等关系数据库、内存数据库、MPP类分布式并行计算数据库等。所以在这里的测试代码中，我计划将输入的铁路图存入H2数据库（使用简单，内置在代码中，无需额外配置）。

表设计如下：



表设计了几个基础字段意思一下。

铁路图数据录入后存入铁路表，再根据铁路表得出城市表，附加字段可以无视。

1. 问题分析
2. 根据给定路线计算距离（路线不存在则提示：NO SUCH ROUTE）

* 根据输入的路线字符串，如：A-B-C。得出经过的城市（A、B、C），通过城市表判断是否经过的城市都在该铁路网中，若有不存在的城市直接返回：NO SUCH ROUTE。
* For循环遍历城市节点，依次取出个节点间的距离，累加得到最终距离，若路线中某两地不存在铁路，则停止计算，返回：NO SUCH ROUTE

1. 给定起点、终点、最大经过站数量给出有多少种路线。

* 先判断给定点是否都在铁路图中，不存在则返回0。
* 使用广度优先搜索，搜索深度为给定深度，很容易得出结果。

1. 给定起点、终点、刚好经过站数量给出有多少种路线。

* 先判断给定点是否都在铁路图中，不存在则返回0。
* 同问题2使用广度优先搜索，结果集少一些而已。

1. 给定起点和终点给出最短路线距离。

* 先判断给定点是否都在铁路图中，不存在则返回：NO SUCH ROUTE。
* 此处应该是使用类似广度优先搜索，但是这里要考虑各分支的距离，避免选中节点少但距离不是最短的路线(如图- 2所示，要避免选中A-F-E)，不能走重复的节点，避免不必要的计算，具体看代码实现。

A

B

F

E

C

D

图- 2

1. 给定起点、终点、小于最大路线距离给出有多少种路线。

* 先判断给定点是否都在铁路图中，不存在则返回0。
* 这个场景可以使用深度或广度优先搜索都行，比较简单。

1. 总结：

总的来看，这个题还是比较简单的，类似大学算法设计中的题，主要就是深度优先搜索和广度优先搜索的运用。

其中问题4相较有复杂一点，其他几个问题解决思路都很简单常规。

我英文不好，代码中全部提示都是中文。

因为示例数据量过小，这里无法测试性能，可能有些算法在数据量大时还需要优化。

测试只编写了问题相关的单元测试，其他衍生出来的工具类等就没有写单元测试了