

Problem Sheet #2

Problem 2.1: *Scotland Yard: Where is Mr. X?*

(10 points)

In the this and following assignments, we will implement a computer program for the board game Scotland Yard. Wikipedia describes the game as follows¹:

Scotland Yard is a board game in which a team of players, as police, cooperate to track down a player controlling a criminal around a board representing the streets of London. It is named after Scotland Yard, the headquarters of London's Metropolitan Police Service. [...] One player controls Mr. X, a criminal whose location is only revealed periodically, and the other players each control a detective, which is always present on the board.

All players start with a number of tokens allowing them to use the following methods of transport:

- Taxis allow the player to move only one space for each token used. They can be used to reach any point in London, most of which are not accessible in this game by other means.
- Buses are available throughout most of the map, allowing longer-distance travel more quickly if the player is located at a bus-stop.
- The London Underground also allows quick travel between distant points of London. However, stations are far apart so the use of an underground ticket can narrow down the possibilities of Mr. X's location.
- Water routes are available, which only Mr. X can use, following the water buses' routes along the Thames between Greenwich and Whitehall.

Each player (Mr. X and the detectives) draws one of 18 possible cards which show where a player has to start, with Mr. X always drawing first. The locations on these cards are spaced far enough apart to ensure that Mr. X cannot be caught in the first round of play. There are a total of 199 locations on the board.

Each detective begins with a total of 22 tokens. Once each transport token is used by a detective, it is turned over to Mr. X, effectively giving him unlimited transport. As he makes his moves, he writes them in a log book or any book and covers them with the tokens he uses, so that the detectives have clues as to his whereabouts. Mr. X also has a number of 'valid on any transport' black tokens equal to the number of detectives in play (in the Milton Bradley version this is always five), and two 'move twice this turn' cards. The water routes require a black token; when one of these is played, the detectives must consider whether or not it is being used to hide a river trip. The detectives must move in the same order each turn so their moves have to be well thought out.

At five specific times during the game, Mr. X has to reveal his current position. Detectives will take this opportunity to refine their search and, if possible, plan ways to encircle him. From each known position, the types of transport used by Mr. X limit the number of possible locations he may be standing in, which provides useful information to detectives (as well as preventing some types of cheating by the fugitive player).

The game is won by the detectives if any of them catches Mr. X by landing on the same square as his current location, or it may be won by Mr. X if he remains out of the grasp of detectives until they all are unable to move (which happens after 22 moves at the latest, since all detectives will have run out of usable tokens by this point).

In this assignment, you will implement a Java program that can read a map of the board. Your program should also be able to answer where Mr. X can be located give his last know position and the list of tokens he has used. A sample execution of a solution is show below:

¹http://en.wikipedia.org/wiki/Scotland_Yard_%28board_game%29

```
$ java ScotlandYardApp london.map 27 taxi taxi bus subway
27 -taxi- 40 -taxi- 52 -bus- 13 -subway- 46
27 -taxi- 40 -taxi- 52 -bus- 13 -subway- 67
27 -taxi- 40 -taxi- 52 -bus- 13 -subway- 89
27 -taxi- 40 -taxi- 52 -bus- 67 -subway- 13
27 -taxi- 40 -taxi- 52 -bus- 67 -subway- 79
27 -taxi- 40 -taxi- 52 -bus- 67 -subway- 89
27 -taxi- 40 -taxi- 52 -bus- 67 -subway- 111
```

The first argument (`london.map`) is the filename with the description of the transportation network (the board). The second argument (27) is the last known position of Mr. X. The remaining arguments present the list of tokens used by Mr. X. The program then prints the possible paths taken by Mr. X.

The map file consists of a sequence of lines. Each line is either an empty line, a comment line (beginning with a `#` character), a line defining a node (beginning with the `node` keyword) or a line defining a typed edge (beginning with one of the keywords `bus`, `taxi`, `ferry`, or `subway`). More precisely, a node line consists of the `node` keyword followed by a natural number identifying a node followed by an X and Y coordinate (both natural numbers). A typed edge line consists of the type name (one of the keywords `bus`, `taxi`, `ferry`, or `subway`) followed by two natural numbers identifying the nodes connected by the typed edge. All items of a line are separated by whitespace characters. Note that nodes must be defined before they can appear in a typed edge.

Below is a simple example map file. The Scotland Yard map file (`london.map`) is linked to the course web page.

```
# example.map --
#

node    1      1      2
node    2      3      4
node    3      5      6

taxi    1      2
taxi    2      3
bus     1      3
```

Some general guidelines:

- Make sure you write clean code and you use Java language constructs in a meaningful way.
- Write a package for the main classes of the game.
- Make sure you are prepared to extend your program in subsequent assignments.
- When designing classes, carefully think about information that should be hidden and information that should be public or shared in the package.
- Try to not expose implementation details if this is not necessary.
- Make sure your program is robust and catches any parsing errors or errors on the command line.