ÇANKAYA UNIVERSITY

CENG 383 Algorithms - Programming Assignment 1

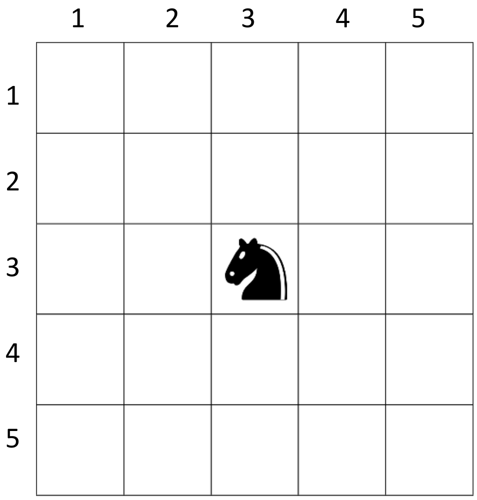
Board

You will be given an $n \times m$ board ($2 \leq n \leq 500$, $2 \leq m \leq 500$). Each cell of the board will be identified using its row and column indices: (i,j). For simplicity, let's name the cell on i^{st} row and j^{th} column as $c_{i,j}$. As you might imagine, this board looks like a 2D array. The top leftmost cell of the board will be $c_{1,1}$ and bottom rightmost cell of the board will be $c_{n,m}$. A sample 3 × 4 board:

$(1,1)$	$(1,2)$	$(1,3)$	$(1,4)$
$(2,1)$	$(2,2)$	$(2,3)$	$(2,4)$
$(3,1)$	$(3,2)$	$(3,3)$	$(3,4)$

Knight

In one of the cells, there will be a knight. There will be **exactly 1 knight on the board**. A sample 5 × 5 board with a knight in $c_{3,3}$:



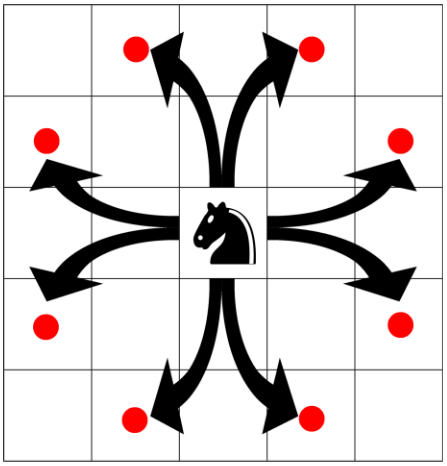
Knight Move

If you are familiar with chess, then you might know what knight moves mean. Don't worry if you are not familiar. You do not have to know the rules of chess to finish this assignment.

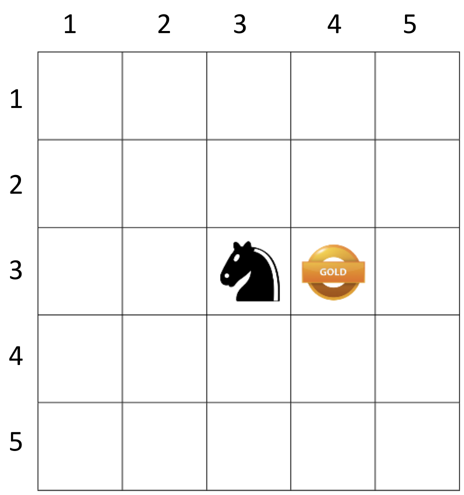
From $c_{i,j}$ the knight can move to one of the following cells (in one step):

$c_{i-2,j+1}$	$c_{i-1,j+2}$	$c_{i+1,j+2}$	$c_{i+2,j+1}$
$c_{i+2,j-1}$	$c_{i+1,j-2}$	$c_{i-1,j-2}$	$c_{i-2,j-1}$

Knight **cannot leave the board.** Possible moves of a given knight are given on the right figure:



Gold



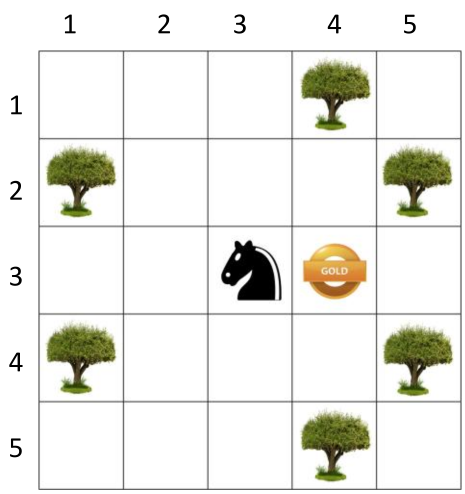
Besides the knight, there will be gold on the board. There will be **exactly 1 gold on the board**. A sample 5 × 5 board is given on right with the knight in $c_{3,3}$ and gold in cell $c_{3,4}$

In **this** assignment, you should find the **length of the shortest path from the initial position of the knight to the cell where gold is located**. In other words, you should find the minimum number of steps for the knight to reach the gold. For the example given the board on the right, a possible shortest path is:

$c_{3,3} \rightarrow c_{2,5} \rightarrow c_{1,3} \rightarrow c_{3,4}$

The length of the shortest path is 3.

Obstacles: Trees



To make the problem slightly harder, we'll insert trees into the board. They will act as an **obstacle**.

A cell with a tree cannot be used by the knight. Therefore, the shortest path should not pass through any cell which contains a tree. The sample board with some trees is given in the right figure:

Now, the knight cannot reach the gold in 3 steps. A possible shortest path from $c_{3,3}$ to $c_{3,4}$:

$c_{3,3} \rightarrow c_{1,2} \rightarrow c_{2,4} \rightarrow c_{3,2} \rightarrow c_{5,3} \rightarrow c_{3,4}$

The length of the shortest path is 5. There isn't any other path which is shorter than 5 steps.

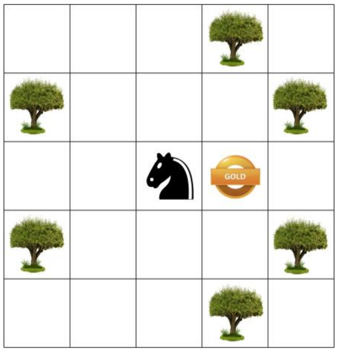
Input / Output Specifications

The board will be given in a string.

- The first 2 characters of the string will be 2 integers: row (r) and column (c) numbers
- Next c characters will be the first row of the board. Other c characters will be the second row and this repeats r times
- "." character shows empty cell
- "K" character is the knight
- "G" character is the gold
- "T" character is a tree
- The text contains **exactly one K** and **exactly one G** characters

The string representation of the board is given below:

$$\texttt{55}\underbrace{\texttt{...T.}}_{\text{1st Row}}\underbrace{\texttt{T...T}}_{\text{2nd Row}}\underbrace{\texttt{..KG.}}_{\text{3rd Row}}\underbrace{\texttt{T...T}}_{\text{4th Row}}\underbrace{\texttt{...T.}}_{\text{5th Row}}$$



The program should read the text as the input and print the path of knight from the initial position to the gold. For example, for the given input board, the result can be:

```
c3,3 -> c1,2 -> c2,4 -> c3,2 -> c5,3 -> c3,4
```

If there is no possible path from knight's initial position to the gold, the program should print:

```
No path to the gold
```

Remarks

- You are **not allowed to use any other external library** for your implementation.
- Generate random boards and test your implementation.
- Your implementation will be tested on different boards. You are recommended to work with smaller boards first and then try on larger ones.

What To Submit

- The Java source code of your program.
- The Java codes should be **WELL FORMATTED** and **WELL DOCUMENTED** as comments, as part of your grade will be based on the level of your comments.

Sample Template to Start

```java
import java.util.*;

public class Main {
   static char[][] board; // Represent the board as a 2D char array

   public static void main(String[] args){
      // First build the board
      Scanner sc = new Scanner(System.in);
      String text = sc.next();
      int row = Character.getNumericValue(text.charAt(0));
      int col = Character.getNumericValue(text.charAt(1));
      …
   }
}
```

- ***HINT:*** Think about the algorithm we used on the last question of slides “Undirected Graphs”, and can you use a similar idea here?

IMPORTANT

1. This assignment is due at 23:59 on Monday, March 24th.

2. You should upload your homework to **WebOnline** before the deadline. No hardcopy submission is needed.

3. The standard rules about late homework submissions apply. Please see the course syllabus for further discussion of the late homework policy as well as academic integrity.

You do not need to prepare a report, just document your code well. You may be asked to explain your code as a face-to-face session. The schedule will be announced later in that case.