CS2010 PS8 - The Toddler Years

Released: Wednesday, 06 November 2013
Due: Saturday, 16 November 2013, 8am

Collaboration Policy. You are encouraged to work with other students or teaching staffs (inside or outside this module) on solving this problem set. However, you **must** write the Java code **by yourself**. In addition, when you write your Java code, you **must** list the names of every collaborator, that is, every other person that you talked to about the problem (even if you only discussed it briefly). This list may include certain posts from fellow students in CS2010 IVLE discussion forum. Any deviation from this policy will be considered cheating, and will be punished severely, including referral to the NUS Board of Discipline. It is not worth it to cheat just to get 15% when you will lose out in the other 85%.

R-option. There is no R-option in this PS.

New Story. After two years plus (24 October 2011 - 06 November 2013), Jane is now no longer a small baby but a *toddler*. As someone who knows (and teaches!) various programming languages, data structures, algorithms, coaches Singapore International Olympiad in Informatics (IOI) + NUS ACM International Collegiate Programming Contest (ICPC) teams, Steven has a hope that Jane likes[1] the field of Computer Science and he will happily teach programming to her, of course free of charge.

However, asking a two years toddler to learn Java (or other high-level programming languages) will be a tall order. The first step is to make Jane interested with programming first. And for that (secondary) purpose, I buy the Lego Mindstorms EV3 robot set (the main purpose is actually for my self entertainment =)...

The Actual Problem. Lego Mindstorms EV3

You have seen the (simplified version) of this problem just now, i.e. during the CS2010 written quiz 2 on Wednesday 06 November 2013, 10.00am-11.17am. The problem statement is replicated here but some statements are edited so that the problem description is consistent from subtask 1 to the new subtask 3. Now, is your Subtask 2 solution during written quiz 2 actually correct? Code it and test it on Mooshak :).

For an extra challenge, I add a new last subtask 3 that you have not seen in the written quiz 2. You do not have to attempt this extra challenge if you are busy with other modules and doing so will only make you end up in level 16 (which does not count).

[1]But if her passion lies somewhere else, e.g. Music, Language, etc, I will not force her to study Computer Science.

Steven has a new toy, LEGO Mindstorms EV3 (which he will pass to Jane in a few years time). He wants to command his robot to move from one cell to another cell in an $M \times N$ grid ($2 \leq M, N \leq 100$) containing mostly '.' (passable cells) and some '#' (blocked cells). The diagram below shows three sample grids of size 4×7, 3×3, and 3×10. There are 8, 4, 8 obstacle cells (the '#'s) in those three sample grids, respectively:

```
Grid 1                  Grid 2          Grid 3
.......                 .##             ..........
.#####.                 .##             .########.
.#...#.                 ...             ..........
...#...
```

Initially, Steven's robot is at coordinate (0, 0)—the top-left corner of the grid—and faces east. At each cell, Steven's robot can only do one of the two actions[2] below:

1. Move forward by one cell according to it's current direction.
   For example, if the robot currently at coordinate (0, 0) and faces east, it will be in coordinate (0, 1) and still faces east after this action.
   Such action consumes 3 seconds.
2. Rotate the robot by 90 degrees clockwise (turn right); the robot stays in the current cell.
   For example, if the robot currently at coordinate (0, 0) and faces east, it will still be in coordinate (0, 0) but now faces south after this action.
   Such action consumes 2 seconds.

New additional feature (not in the written quiz 2): For those who are familiar with robots, instructing a robot to turn by 90 degrees clockwise usually will <u>not</u> yield an accurate 90 degrees clockwise turn due to friction with floor (and various other reasons). We may end up with a 89.5 degrees clockwise turn. At another time, we may end up with 92.1 degrees clockwise turn, etc.

Write a program that Steven can upload to his robot so that his robot can move from coordinate (0, 0) to coordinate (A, B) using the **minimum amount of time** and using **no more than K right turns, i.e. the action 2 above** ($0 \leq A < M; 0 \leq B < N; 0 \leq K \leq 100$)! Of course, Steven's robot **cannot** go outside the $M \times N$ grid and cannot move to a blocked cell[3] throughout the execution of the program. When the robot stops at coordinate (A, B), it can face any direction. Output this minimum time. However, if there is no way to reach coordinate (A, B) using no more than K right turns, output -1.

Subtask 1/Problem A in Mooshak (30 points). You are given a simplifying special case. The action 2 (turn right) consumes 0 second, i.e. 'it is instantaneous' and it is guaranteed that you can ignore parameter K (it will be very large, 1 billion for this subtask). For the three sample grids with various A, B, and K settings, the answers are 27, 14, and 9 seconds, as shown below:

```
Grid 1                      Grid 2                  Grid 3
(A,B) = (3,6)               (A,B) = (2,2)           (A,B) = (2,1)
K = 1000000000              K = 1000000000          K = 1000000000

    111                     0##                     0.........
0369258                     3##                     3########.
.#####1<-(actually 21)      692<-(actually 12)      69........
.#...#4<-(actually 24)                               ^-(nine)
...#..7<-(actually 27)
```

[2]For your information, there is one official robot model in the Lego Mindstorms EV3 retail set that behaves like this. Please visit http://algorithmics.comp.nus.edu.sg/mediawiki/index.php/LegoNUS#SPIK3R if you are interested.

[3]In case you ask, the two cells (0, 0) and (A, B) will never be blocked.

Subtask 2/Problem B in Mooshak (50 points). This time, the action 2 (turn right) consumes 2 seconds as mentioned in the problem description. For this subtask, it is guaranteed that you can also ignore parameter K (it will be very large, 1 billion for this subtask). For the three sample grids with various A, B, and K settings, the answers are 29, 20, and 17 seconds, as shown below:

```
Grid 1                      Grid 2                  Grid 3
(A,B) = (3,6)               (A,B) = (2,2)           (A,B) = (2,1)
K = 1000000000              K = 1000000000          K = 1000000000

    111
0369258                     0##                     0.........
.#####3<-(actually 23)      5##                     5########.
.#...#6<-(actually 26)      870<-(actually 20)      87........
...#..9<-(actually 29)       ^---(actually 17)       ^---(actually 17)
```

Subtask 3/Problem C in Mooshak (20 points). This is the full version. The action 2 (turn right) consumes 2 seconds. You cannot ignore parameter K but it will have reasonable range, i.e. $0 \leq K \leq 100$. For the three sample grids with various A, B, and K settings, the answers are 29, -1, and 61 seconds, as shown below:

```
Grid 1                      Grid 2                  Grid 3
(A,B) = (3,6)               (A,B) = (2,2)           (A,B) = (2,1)
K = 1                       K = 3                   K = 2

    111                                                 111222
0369258                     0##                     0369258147
.#####3<-(actually 23)      5##                     .########2<-(actually 32)
.#...#6<-(actually 26)      8..                     .185296305<-(actually 35)
...#..9<-(actually 29)      ^-stuck, 4 turns         ^---(actually 61)
                              output -1
```

Explanation:

On the sample grid 1 above, the best robot path from (0, 0) to (3, 6) only uses 1 right turn, so the answer is the same as in Subtask 2, which is 29 seconds.

On the sample grid 2 above, there is no robot path from (0, 0) to (2, 2) that uses no more than 3 right turns (the robot needs at least 4 right turns). Therefore, we output -1.

On the sample grid 3 above, the best robot path from (0, 0) to (2, 1) that uses no more than 2 right turns is the **detour path**: move forward 9 times (27 seconds), turn right once at coordinate (0, 9) (2 seconds), move forward 2 times (6 seconds), turn right once at coordinate (2, 9) (2 seconds), and finally move forward 8 times (24 seconds) to reach coordinate (2, 1). As this path only uses 2 right turns, it is a valid path and uses 27+2+6+2+24 = 61 seconds. The 17 seconds solution from Subtask 2 is invalid for this Subtask 3 as it uses 4 right turns.

Note: The official test data has been uploaded to Mooshak online judge, but it is hidden from your view. The time limit setting in Mooshak online judge for Subtask 1,2,3 are 1,1,1 seconds, respectively. You are encouraged to generate and post additional test data in Facebook group/IVLE discussion forum. You are only given `Sample1.txt`, `Sample2.txt` (technically the same as `Sample1.txt`), `Sample3.txt`, `Sample1-ans.txt`, `Sample2-ans.txt`, and `Sample3-ans.txt` that correspond to the samples shown above.

This is the end of CS2010 Problem Sets.
If you want more, consider taking CS3233 - Competitive Programming module =).