CSCI 3110: Warm-Up Assignment (3% of final mark) posted: May 6th, 2024
Instructor: Travis Gagie due: 23:59 May 12th, 2024

Unlike the other assignments in this course, you are to do this assignment **by yourself**. It should take you at most a few hours. If you need to cheat by consulting friends / the internet / ChatGPT / whatever then you are probably not ready for this course and you should seriously consider **dropping it while you can still get your money back**.[1] Of course the choice to drop is yours but although your friends / the internet / ChatGPT / whatever may be available to help you cheat on all the other assignments as well, they will not be available to help you during the exams and **if you fail the final exam then you fail the course**.[2]

This assignment is due Sunday night at 23:59, Halifax time. There are no SDAs for this course. Late submissions will not be accepted unless you have an accommodation through the Accessibility Center that allows you to submit work up to 3 days late, in which case we will accept your assignment until 23:59 on Wednesday.

1. Suppose you have a little puzzle consisting of 8 square tiles numbered 1 to 8 on a 3 × 3 grid, with one grid cell left empty. Initially the tiles are arranged with 1, 2 and 3 in that order in the top row, 4, 5 and 6 in that order in the second row, and 7, 8 and the empty cell in that order in the bottom row. You are allowed to slide any tile that is adjacent to the empty cell horizontally or vertically (not diagonally) into the empty cell; this counts as 1 move.

   You are to write a program that reads from `stdin` a configuration of the puzzle and writes to `stdout` the minimum number of moves needed to reach that configuration from the initial configuration (or `unreachable` if the given configuration is unreachable). The configuration will be given as 3 lines of text with 3 symbols in each line, separated by spaces; the empty cell will be indicated with a `0`. For example, on the inputs

```
1 2 3        1 2 3        3 5 8        3 5 8
4 5 6        4 0 5        0 1 2        0 1 2
7 8 0        7 8 6        4 6 7        4 7 6
```

   your program should output `0`, `2`, `17` and `unreachable`, respectively.

   You can download a file with 1000 inputs from the BrightSpace page. Submit your source code and a text (`.txt`) file with your program's 1000 corresponding outputs, one per line. You will get all 3 points if *all* your outputs are correct and 2 if you clearly know what's going on but your code is a bit buggy (and anything below that is a bad sign).

   You should use breadth-first search on the graph whose vertices are the configurations of the puzzle with an edge between two vertices if those configurations are reachable from each other with 1 move. Recall that breadth-first search means using a queue and some means (such as a hash table) to check whether you've visited a vertex before. You can use any programming language you like as long as the head TAs say it's ok. It may take them a day or two to check with that the markers know the language. If you don't want to wait then use C, C++, Java or Python. (I use C myself but the libraries in C++, Java or Python would probably make things easier.)

[1] Personally I think that if you can't do this assignment then you should get your money back for CSCI 2110 as well (but good luck with that).

[2] I haven't enforced that rule very strictly before but given how many students are using ChatGPT now, I don't think I have a choice anymore.