CMPT 310 - Artificial Intelligence Survey

Assignment 2

Due date: October 1, 2018 | J.P. Delgrande
10 marks | September 14, 2018

Important Note: Students must work individually on this, and other CMPT 310, assignments. You may not discuss the specific questions in this assignment, nor their solutions with any other student. You may not provide or use any solution, in whole or in part, to or by another student.

You are encouraged to discuss the general concepts involved in the questions in the context of completely different problems. If you are in doubt as to what constitutes acceptable discussion, please ask!

This problem involves a highly simplified version of path-finding taking terrain into account. You are given an $n \times n$ grid (with co-ordinates in the range $0, \ldots, (n-1)$), a starting position on the grid and a goal location. Each point on the grid is assigned an integer, giving its elevation. It is possible to move from a location to any of its 4 neighbours (except at a boundary). The cost of such a move in 1 + the absolute value of the difference in elevation. However, if the difference between two points is ≥ 4, this represents an impassible cliff: the agent can't go "up" a cliff (i.e. move from the lower-ranked point to the higher), and going "down" will wreck the agent. The goal is to determine the least cost path from the start location to the goal location. The following is an example; conventionally the top left hand corner is $(0, 0)$ and the lower right corner is $(n-1, n-1)$.

```
[[1 1 3 2 2]
 [2 2 3 2 2]
 [2 3 6 7 2]
 [3 4 5 3 2]
 [2 4 5 4 2]
 [2 3 4 2 2]]
```

Write a program in Python that reads in a problem specification in the folowing order: the grid size, the start location, the goal location, and the array of values. The program should determine

- the result found by best-first search, and
- the result found by A^* search.

A test case is provided separately. You should submit the following:

- Your program.

- Testing. This will include the test case, together with any other testing that you feel illustrates what your program does.

  Output should include the number of nodes expanded, the path found, along with anything else that you feel may be helpful.

- Documentation. This will include a description of the heuristic that you choose. As well, the documentation should describe your program at a high level. If you did anything particularly nice, please document that also.