COMP3411/9814 Artificial Intelligence

Term 1, 2024

Assignment 1 – Bridge Puzzle

Due: Friday 15 March, 10pm
Marks: 12% of final assessment

Specification

This project is based on a popular puzzle, variously known as "Hashiwokakero", "Hashi" or "Bridges". You will need to write a program to solve this puzzle, and provide a brief description of the algorithm and data structures you have used. The input to your program will be a rectangular array of numbers and dots, for example:

```
. 1. . . 6. . . 7. . . . 4. 4. 2.
. . 4. 2. . 2. . . 3. 8. . . 6. 2
. . . . . 2. . . . . . . . . . . .
5. c. 7. . a. a. . 5. 6. . 8. 5
. . . . . . . . . . . . 2. . . . . .
. . . 5. . . 9. a. . 8. b. 8. 4.
4. 5. . . . . . . . . . . . . . . 3
. . . . 2. . 4. . 1. 5. . . 2. . .
. 2. 7. 4. . . 7. 2. . 5. . . 3.
. . . . . . . . . . . . 4. . 3. 1. 2
```

Each number represents an "island", while the dots represent the empty space (water) between the islands. Numbers larger than 9 are indicated by 'a' (10), 'b' (11) or 'c' (12). The aim is to connect all the islands with a network of bridges, satisfying these rules:

1. all bridges must run horizontally or vertically
2. bridges are not allowed to cross each other, or other islands
3. there can be no more than three bridges connecting any pair of islands
4. the total number of bridges connected to each island must be equal to the number on the island

For example, after reading the 10-line input above, your program might produce this output:

```
1---6EEE7====4=4=2
 4-2″ 2 ″ 3E8EEE6 2
 # |2 ″ ″   ″   # ″
5EcE7EEaEa==5″6EE8=5
″ #    ″ #  #2#    |
″ #5===9Ea--8=bE8E4|
4=5#   ″ #  ″ # ″ |3
   #2==4 #1-5 # 2 |″
 2=7=4===7=2″ 5===3″
            4==3-1 2
```

Note that single bridges are indicated by the characters '-' or '|', pairs of bridges by '=' or '″' and triples by 'E' or '#', depending on whether they run horizontally or vertically. Water between bridges and islands is indicated by space characters ' '.
In some cases, there may be many solutions, in which case your program should only print one solution. More details about the puzzle can be found on this Wikipedia page. Note, however, that our version allows up to 3 bridges instead of 2; also, we do not insist that the entire graph be connected.

Tools

An executable file called `bridgen` is provided in the `tools` directory which can be used to generate sample data of any specified size (type `bridgen -help` for details). Another executable called `bridgecheck` is also provided, to help you test the validity of your solutions (see FAQ for details).

Questions

At the top of your code, in a block of comments, you must provide a *brief* answer (one or two paragraphs) to this Question:

> Briefly describe how your program works, including any algorithms and data structures employed, and explain any design decisions you made along the way.

Language Options

You are free to write the code in a language of your choosing.

- If you write in C, C++, or another compiled languge, your program will be invoked by: `./hashi`

You should submit your source files (no object files) as well as a `Makefile` which, when invoked with the command `make`, will produce an executable called `hashi`

- If you write in Python, your program will be invoked by: `./hashi.py`
  You should submit your `.py` files (including `hashi.py`).
  The first line of your code must specify which version of Python you are using, e.g. `#!/usr/bin/python3`
- If you write in Java, your program will be invoked by: `java Hashi`
  You should submit your `.java` files (no `.class` files).
  The main file must be called `Hashi.java`
- If you wish to write in some language not covered by the above options, let us know and we will try to accommodate you.
- Regardless of the language, you are not allowed to use dedicated constraint programming packages like `python-constraint`, etc. You are expected to implement the search method(s) yourself.

Submission

You should submit by typing

```
give cs3411 hashi ...
```

Remember to include all necessary files in your submission (including the one with the answer to the Question).

You can submit as many times as you like – later submissions will overwrite earlier ones. You can check that your submission has been received by using the following command:

```
3411 classrun -check
```

The submission deadline is Friday 15 March, 10 pm.
5% penalty will be applied to the mark for every 24 hours late after the deadline, up to a maximum of 5 days (in accordance with UNSW policy).

Additional information may be found in the FAQ and will be considered as part of the specification for the project.

Questions relating to the project can also be posted to the Forums on WebCMS.

If you have a question that has not already been answered on the FAQ or the Forums, you can email it to `cs3411@cse.unsw.edu.au`

Assessment

Your program will be tested on a series of sample inputs of successively increasing size and difficulty. There will be:

- 6 marks for functionality (automarking)

- 4 marks for your algorithm and implementation
- 2 marks for answer to the Question

You should always adhere to good coding practices and style. In general, a program that attempts a substantial part of the job but does that part correctly will receive more marks than one attempting to do the entire job but with many errors.

Groups

This assignment may be done individually, or in groups of two students. Groups are determined by an SMS field called `pair1`. Every student has initially been assigned a unique `pair1` which is `"h"` followed by their student ID number, e.g. `h1234567`.

1. If you plan to complete the assignment individually, you don't need to do anything (but, if you do create a group with only you as a member, that's ok too).
2. If you wish to team up with someone, you should go to the WebCMS page and click on "Groups" in the left hand column, then click "Create". Click on the menu for "Group Type" and select "pair". After creating a group, click "Edit", search for the other member, and click "Add". WebCMS assigns a unique group ID to each group, in the form of `"g"` followed by six digits (e.g. `g012345`). We will periodically run a script to load these values into SMS.

Plagiarism Policy

Your program must be entirely your own work. In addition, soliciting another person (or an AI bot) to write code for you – either in person or through the Internet – is never permitted. Generally, the copying of code already available on the Internet is also forbidden. If you find some piece of "standard" code in a textbook, or on the Internet, which you would like to adapt and incorporate into your own assignment, you must email the lecturer in charge to ask if it is permissible to do so in the particular circumstances – in which case the source would have to be acknowledged in your submission, and you would need to demonstrate that you had done a substantial amount of work for the assignment yourself. Plagiarism detection software will be used to compare all submissions pairwise and serious penalties will be applied, particularly in the case of repeat offences.

DO NOT COPY FROM OTHERS; DO NOT ALLOW ANYONE TO SEE YOUR CODE

Please refer to the UNSW Policy on Academic Integrity and Plagiarism if you require further clarification on this matter.

Good luck!
