COMP1039 Coursework 2 (15 Marks)
Release Date: 8 April 2024 18:00
Deadline: 6 May 2024 18:00

Your Task

A farmer with his wolf, goat, and cabbage come to the edge of a river they wish to cross. There is a boat at the river's edge, but, of course, only the farmer can row it. The boat also can carry only two things (including the rower) at a time. If the wolf is ever left alone with the goat, the wolf will eat the goat; similarly, if the goat is left alone with the cabbage, the goat will eat the cabbage. Implement the problem in Haskell, and use a search algorithm to find the solution.

Suppose that the function is called `solutionPath`. The initial state is `['w', 'w', 'w', 'w']`, in which the four w's represent the position of the farmer, wolf, goat, and cabbage respectively, at the west bank of the river. The final state is `['e', 'e', 'e', 'e']` in which the four e's represent the position of the farmer, wolf, goat, and cabbage respectively, at the east bank of the river.

We can apply the function to find the solution path as follows:

```
solutionPath [['w', 'w', 'w', 'w'] ['e', 'e', 'e', 'e']]
```

One possible returned solution path is as follows:

```
[['w', 'w', 'w', 'w'], ['e', 'w', 'e', 'w'], ['w', 'w', 'e', 'w'],
['e', 'e', 'e', 'w'], ['w', 'e', 'w', 'w'], ['e', 'e', 'w', 'e'],
['w', 'e', 'w', 'e'], ['e', 'e', 'e', 'e']]
```

The solution path is interpreted as follows:

[F, W, G, C]	[Farmer, Wolf, Goat, Cabbage]
['w', 'w', 'w', 'w']	Farmer, Wolf, Goat, and Cabbage are at the west bank of the river
['e', 'w', 'e', 'w']	Farmer takes the Goat across to the east bank
['w', 'w', 'e', 'w']	Farmer returns to the west bank alone
['e', 'e', 'e', 'w']	Farmer takes the Wolf across to the east bank
['w', 'e', 'w', 'w']	Farmer returns with the Goat to the west bank
['e', 'e', 'w', 'e']	Farmer takes the Cabbage across to the east bank
['w', 'e', 'w', 'e']	Farmer returns alone to the west bank
['e', 'e', 'e', 'e']	Farmer takes the Goat across to the east bank
	Finally, all four of them are at the east bank.

Your Submission

Please ensure that your report is formatted in Microsoft Word and contains the following sections:

Section A: Solving the Farmer Crosses River Puzzle using State Space Search (4 marks)
In this section, provide an explanation in no more than 500 words on how you can solve the Farmer Crosses River puzzle using the state space search problem-solving technique. Utilize a figure to illustrate clearly how your search algorithm traverses either a tree or a graph.

Section B: Haskell Source Code (4 marks)
Include your Haskell program without any remarks/comments. Your program will be evaluated based on the following criteria:

- Correctness: The program should produce the required output with correct input.
- Conciseness: Aim for brevity while ensuring the program meets the required specifications.
- Adherence to Functional Programming Practices: The code should follow good functional programming practices.
- Utilization of List Structure: Represent the problem states using list structure ([F, W, G, C] as demonstrated).

Section C: Experimentation with the Program (Input-Output Sessions) (4 marks)
Include up to 5 screenshots showcasing the output of your experiments. These screenshots should depict the solution paths generated when given initial and final states as input. Your program will be assessed based on the following features:

- Ability to generate all possible solution paths, representing safe ways of crossing the river.
- Effective functionality even when one or two of the wolf, goat, or cabbage are already on the opposite side of the river.
- Identification of the shortest solution path.
- Ability to inform the number of trips required to get everything across the river safely.

Section D: Discussion on Solving the Same Problem in OOP Way (3 marks)
Provide an explanation in no more than 300 words on how you can solve the Farmer Crosses River puzzle in Object-Oriented Programming way using Java. You should compare and contrast them in terms of programming styles (not syntax), such as inheritance, polymorphism, function overloading, as well as the imperative and declarative nature of the paradigms.

Submission Instructions

There are two files to be submitted: a report in Microsoft Word (e.g., Report.docx), and a Haskell program source file (e.g., Farmer.hs). Please ensure that all your files are compressed into a single zip file. The file should be named according to the following format: STUDENTID_NAME.zip (e.g., 20514000_Danting_Wang.zip).

Once your files are organized and zipped, submit the zip file onto the Moodle page. Note that each subsequent submission will overwrite the previous one. If you submit multiple times, please verify that your last submission includes all the necessary files.

After submission, please review your submission to ensure that it is complete and executable. Past experiences have shown that submitted files may occasionally be corrupted. You will receive a zero mark if your submitted file is corrupted or not executable.

For late submissions, the standard late submission policy applies, resulting in a 5% deduction for every 24-hour period, including weekends and public holidays.

Plagiarism

If you utilize code sourced from a textbook or the internet, you must acknowledge its origin. Plagiarism detection tools will be employed to check for similarities between submissions and the online content. We would like to remind you of the School's Policy on Plagiarism. While recent advancements in AI and large language models (LLMs) such as ChatGPT have presented good opportunities of using them, it is crucial to note that the University considers the use of such technologies as potentially constituting misconduct.

Recommended Learning Materials

Welcome to the Farmer, Wolf, Goat, and Cabbage Problem
https://www.d.umn.edu/~gshute/cs2511/slides/javascript/code/solve/solve.html
There is a simple animation to help you understand the Farmer Crosses River puzzle better, before you start to implement a solution.

AI Algorithms, Data Structures, and Idioms in Prolog, Lisp, and Java
https://www.cse.sc.edu/~mgv/csce580sp15/Luger_0136070477_1.pdf
There is a good discussion about implementing the Farmer Crosses River puzzle using Prolog (logic programming), Lisp (functional programming), and Java (object-oriented programming).

Graph theory: 🦊 wolf, 🐑 sheep and 🥬 cabbage
https://www.youtube.com/watch?v=pBT-8gqhHzo
This videoclip provides a simple and easy to understand presentation on how to solve the Farmer Crosses River puzzle using a graph search method.

Goat, cabbage, wolf
https://www.uni-weimar.de/fileadmin/user/fak/medien/professuren/Webis/teaching/ws14/search-algorithms/wolf-goat-cabbage.html
There is a good discussion on how to formulate the Farmer Crosses River puzzle as a search problem.

A little bit of Lisp
https://cse.unl.edu/~choueiry/F23-476-876/Slides/lisp.pdf
There is a discussion about the Farmer Crosses River puzzle in Lisp.