CS7IS2: Artificial Intelligence Assignment 1

In this assignment you will implement a number of search and MDP algorithms and compare their performance in a maze search. This assignment is worth **35% of your overall CS7IS2** mark.

Please note this is an **individual** assignment. You are allowed to discuss ideas with your colleagues but do not share code. Any suspected plagiarism will be dealt with in line with University policies https://www.tcd.ie/teaching-learning/academic-policies/plagiarism/

Submissions are via Blackboard, and due by **Friday February 28th 5pm**.

You can **penalty-free submit assignment until Sunday March 2nd** evening (midnight) but please note there will be no Blackboard/email/TA/lecturer support past the official deadline. Requests for extensions (only on medical or other serious grounds) have to be received before **February 28th 5pm**.

Assignments submitted later than Sunday March 2nd will be **marked down by 33% of the achieved mark per day (**or a part of the day – ie assignments submitted any time on March 3rd will all be reduced by 33%, on March 4th by 66%, and by March 5th by 99%, ie effectively no marks will be awarded).

Assignment specification:

1. Implement in Python (or reuse existing open source, with appropriate referencing) maze generator, capable of generating mazes of varying sizes. (Please note reuse applies only to maze – remainder of the assignment must be your own work/own intellectual property. TCD policies on plagiarism and use of GenAI apply)
2. Implement search algorithms DFS, BFS, and A* for maze solving
3. Implement MDP value iteration and policy iteration algorithms for maze solving
4. Compare the performance of all 5 algorithms from points 2 and 3 to each other in a range of mazes of different sizes and using multiple metrics. Analyse and discuss (1) the comparison between different search algorithms to each other, (2) comparison between different MDP algorithms to each other, (3) comparison between search and MDP algorithms.

   Deliverables: Python code, raw data from experimental analysis (eg csv file), a report **analysing and discussing the performance comparisons**, including visual comparison/graphs/tables/screenshots as required. This document also needs to include **justification for any design choices** you make in the implementation (e.g., choice of a heuristic, choice and impact of MDP parameters used, choice of comparison metrics etc.)
5. Record a brief (max 5 minutes) video demo (screen grab + voiceover) of your maze solvers (showing input parameters, output screens etc)

Submission instructions

Please submit a **single zip file** containing all python code, **readme.txt** with command lines use to run each of your algorithms, **pdf of the performance analysis document**, results files, and the **video** of the demo. In addition, code for the functions containing implementations of the 5 algorithms **need to be added as the appendices 1-5 to the document**. Please note the maximum file size for the

whole submission is 100mb, so you'll need to lower the quality of your demo video if it defaults to more than this (or include only a link to a publicly accessible file).

Additional Important Notes:

- If reusing existing open source implementation of a maze generator, **make sure to credit it in your design document (including link to the repository).** Make sure that the licence under which the code is distributed allows you to reuse it. Similarly, if reusing any snippets of code for the algorithms themselves, **make sure they were fully credited both in your code, as well as in the report.**

- While no marks are awarded specifically for the demo, the demo is a **mandatory** part of the assignment, without which the rest of the assignment will not be marked

- The sizes of mazes for which to analyze the performance are not pre-specified; it is up to you to select suitable sizes such that execution times for your algorithms are feasible, but such **that the difference in performance between algorithms is noticeable**. If it takes some trial and error to identify suitable sizes of the maze, please include this in the design document, outlining which other sizes have you tried and why they were not suitable for inclusion/analysis. Be patient – larger sizes will take longer to run!

- Similarly, the metrics which to use to analyze the performance are not pre-specified, it is up to you to identify suitable metrics for each algorithm/algorithm family, and in the design document to justify your choices.

- Note that, while assignment is asking to report only on several sizes of the maze, your code should be able to run on any maze size/type, ie mazes should not be hard-coded.

- The purpose of the demo is to demonstrate that your code is fully functional and meets the brief requirements, in case we are unable to execute your source files due to compatibility issues. Please ensure that the video sufficiently captures this.

- You are not allowed use any GenAI-generated text in the report. Feel free to use GenAI to enhance your understanding of the topics and algorithms, but any discussion in the report needs to be written by you and refer to specifics of your implementations and linked to your own results and graphs, rather than be generic GenAI content! TCD Policies on plagiarism and use of GenAI apply.