

## 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 **25% 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 March 10th 5pm**.

You have an option to **penalty-free submit assignment until Sunday March 12<sup>th</sup>** evening (midnight) but please note there will be no Blackboard/email/TA/lecturer support for assignment or any other technical issues past the official deadline. Any requests for extensions (on medical or other similarly serious grounds) have to be received before the official deadline.

Assignments submitted later than Sunday 12th will be **lowered by 20% of the achieved mark per day**.

### Assignment specification:

1. Implement in Python (or reuse existing open source) maze generator, capable of generating mazes of varying sizes
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 at least 3 different sizes using a range of at least 2 suitable 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, document **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) screen-grab (and if required voiceover) video demo of your maze solvers (showing input parameters, output screens etc)

### Mark breakdown

1. 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
2. Algorithm implementation – 10% per algorithm, for the total of 50% of the mark

3. Analysis document 50% (proportional to algorithms implemented, eg if only 3 of the above algorithms are implemented and compared, then the maximum achievable document mark is 30%).

### 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**, 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.
- 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 briefly in the design document, outlining which other sizes have you tried and why they were not suitable for inclusion/analysis.
- 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.
- 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.