

Artificial Intelligence

Project 1: Search

Deadline: October 27, 2024 at 23:59

General Remarks

- You are free to consult with other students. The assignment is, however, individual, meaning that each submitted assignment has to be sufficiently unique.
- Always motivate your answers. Detail the steps needed to come to your conclusions.
- You are free to provide answers in either English or Dutch.
- Solutions should work with Python version 3.8.

(1) (Un)Informed Search (14 points)

The code and background information can be found [here](#). Solve the following questions in their respective order:

Question 1 (2 points): Finding a Fixed Food Dot using Depth First Search

Question 5 (4 points): Finding All the Corners

Question 6 (3 points): Corners Problem: Heuristic

Question 7 (5 points): Eating All The Dots

Report (3 points): Discuss Heuristics

- Describe your heuristics for questions 6 and 7 and discuss their *consistency* and *admissibility* (see `report.md`). You do not need to give a formal proof, but rather try to use intuitive arguments.

Grading of questions 6 and 7 depend on the strength of your heuristic. Remark that the number of expanded nodes for questions 6 and 7 is counted whenever you call the `problem.getSuccessors()` function. If you bypass this function this constitutes cheating.

Your heuristic must be a non-trivial, non-negative, consistent heuristic to receive any points. Note that, even if the autograder does not detect a situation where your heuristic is inconsistent, this does not prove consistency.

(2) Adversarial Search (5 points)

The code and background information can be found [here](#). Please download the entire code again, and do not use files from the previous section. Solve the following questions:

Question 3 (3 points): Alpha-Beta Pruning

Question 4 (2 points): Expectimax

(3) Constraint Satisfaction Problems (15 points)

The code and background information can be found [here](#). Please download the entire code again, and do not use files from the previous section. Solve the following questions:

Question 3 (4 points): Forward Checking (part 2)

Question 4 (3 points): Sudoku

Report (1 points): Forward Checking

- Report on the average amount of calls required for $n=50$ with and without `CSP::selectVariable` and `CSP::orderDomain` (you can use the flags `--no-mrv` and `--no-lcv`). Can you explain why these two functions make the algorithm faster/slower (see 'report.md')?

(4) Report

You are provided with a template document, `report.md`, in markdown format for submitting the additional questions. Replace the square brackets with your answers, and ensure that your responses are concise and to the point.

Submission

Upload a **zip** file containing **only** the files:

(1) (Un)Informed Search:

- `search.py`
- `searchAgents.py`

(2) Adversarial Search:

- `multiAgents.py`

(3) CSP:

- `CSP.py`
- `Sudoku.py`

(4) `report.md`

These files should contain your solution to all the questions. Do not change the names of these files.

Good Luck!
