

CS 3346A / CS 3121A Assignment 1: Search

Due: Friday, Oct. 18th, 2019 (midnight)

Silent Policy: A silent policy will take effect 24 hours before this assignment is due, i.e. no question about this assignment will be answered, whether it is asked on the discussion board, via email or in person.

You can either choose Option 1 (from Berkeley U) or Option 2 (from U of Toronto). Permission for reuse has been obtained. This assignment must be done with individual effort (no group work).

Option 1 : Pacman Project: Search

Weight: 15%

Please check the [Berkeley's cs188's Project 1's page](#) for detailed info and instruction. Please download the code files [search_pacman.zip file here](#).

You will essentially finish questions 1 to 7 in this project, except we have a different weighting: Q1 to Q6, 2 points each; Q7, 3 points - for a total of 15 points)

Submission

Please submit your final implementation python files: `search.py` and `searchAgents.py` through OWL.

Please do not change the other files in this distribution or submit any of our original files other than these files.

Evaluation: Same as Berkeley's Project 1.

If necessary, we will review and grade assignments individually to ensure that you receive due credit for your work. Please add concrete **comments** on your implemented functions to show your understanding.

As described in the course outline, we will use plagiarism-checking software. If you copy someone else's code and submit it with minor changes, we will know.

Note :

If you need to pick up python and Unix basis, please learn this

[Unix/Python/Autograder Tutorial](#).

This assignment assumes you use **Python 3.6** or above. Please make certain that your code runs on **python3** using only standard imports.

Option 2 : Sokoban Project

Weight: 16%

(This option is for those who are interested in the Sokoban problem, and are willing to learn more deeply algorithms beyond the lecture. Students who choose this option will have a chance to earn an additional **1%** bonus mark.)



Please check the [UoT csc384's Sokoban introduction](#) for problem description, codes and algorithms introductions.

Please download the code file [search_sokoban.zip](#) here.

Goal (same as U of T but with our own weighting for a total of 16 points):

You will essentially finish the following functions in **solution.py**:

- **Manhattan distance heuristic** (*heur manhattan distance(state)*) **and non-trivial heuristic search** (*heur alternate(state)*). (4 points)
- **Anytime Greedy Best-First Search**. (5 points)

Details regarding this algorithm are provided in [Section 5](#).

- **Anytime Weighted A *** (5 points)

Details about this algorithm are provided in [Section 6](#).

- **Five tips** (2 sentences each) as if you were advising someone who was attempting this problem for this first time on what to do. Write these tips in the file `tips.txt` (2 points)

Submission

Submit your modified `solution.py` and `tips.txt` through OWL.

Please *do not* change the other files in this distribution or submit any of our original files other than these files.

Evaluation: Same as UoT's assignment

If necessary, we will review and grade assignments individually to ensure that you receive due credit for your work. Please add concrete **comments** on your implemented functions to show your understanding.

This assignment assumes you use **Python 3.7** or above. Please make certain that your code runs on **python3** using only standard imports.

As described in the course outline, we will use plagiarism-checking software. If you copy someone else's code and submit it with minor changes, we will know.