COMP30024 Artificial Intelligence

Project Part A
Submission Instructions

Sarah Erfani, Matt Farrugia, Chris Leckie
Last updated: 24 March 2018

Submission

One submission is required from each group. That is, one group member is responsible for submitting all of the necessary files that make up your group's solution.

You must submit a single compressed archive file (e.g. a `.zip` or `.tar.gz` file) containing all files making up your submission via the 'Project Part A Submission' item in the 'Assessments' section of the LMS.

This compressed file should contain **all Python files** required to run your program, **and your comments.txt** file too.

Running your program

The entry point to your program should be a Python file called '**`parta.py`**' (case sensitive). Note that you may structure your program over multiple Python files, but we will look for a file called `parta.py` when testing your program.

When marking your program, we will run your program with a command of the form:

```
python parta.py < input.txt
```

where

- `python` is the name of a Python 3.6 interpreter, and
- `input.txt` is the name of a file containing a board configuration and a command, as per the specification for Project Part A.

We will test your program on the Melbourne School of Engineering Student Unix machines (e.g. `dimefox.eng.unimelb.edu.au` or `nutmeg.eng.unimelb.edu.au`). If you develop your program with a different version of Python, it is your responsibility to make sure it runs correctly using Python 3.6 on the Student Unix machines.

Test cases

Please download the following input files and their corresponding sample output files. You can use these to test the format of your program's output. Keep in mind that your program will be tested on more inputs after submission, and that for some 'Massacre' inputs there may be multiple correct output sequences meaning your output might be different from the output provided. You don't need to worry about having a different (but correct) output sequence as long as you make sure your output is formatted correctly.

Input file	Sample output file
`move-sample-1.in`	`move-sample-1.out`
`move-sample-2.in`	`move-sample-2.out`
`move-sample-3.in`	`move-sample-3.out`
`massacre-sample-1.in`	`massacre-sample-1.out`
`massacre-sample-2.in`	`massacre-sample-2.out`
`massacre-sample-3.in`	`massacre-sample-3.out`

For example, to test your program against the first 'Massacre' sample (massacre-sample-1), you would run the following command:

```
python parta.py < massacre-sample-1.in
```

and then compare the output of your program to `massacre-sample-1.out`.

Warning: These files contain Unix-style newlines. **They will not display properly in some text editors, including the default Windows text editor Notepad.**

Assessment

Please see the original specification partA-spec-2018.pdf for details on how Project Part A will be assessed.

In addition, please note the following:

- For Project Part A, your program must only use tools from the Python Standard Library. You may not use third-party packages. In particular, you must not use `numpy` in your program (even though it is available under dimefox's Python 3).
- Your program must not use multithreading (even though the tools for writing multi-threaded Python programs are part of the Python Standard Library).

Finally, please note that for the tests we use to assess your program in calculating 'Massacre', your program will be subject to a time limit and may be terminated before finishing its computation if it is taking too long.
In order to guarantee that you are able to pass all of our tests (assuming you have an otherwise correct implementation), you should ensure that your program is able to find move sequences involving **up to five moves (total)** for **up to five White pieces** in **under 30 seconds** on dimefox. Note also that our largest tests will represent only a small portion of all tests, most tests will be significantly smaller.

Submission

The submission deadline for Part A is 4.00pm Friday 30th March 2018.

Late submission

Late submissions will incur a penalty of two marks per working day (or part thereof).

If you cannot submit on time you should contact both Sarah (sarah.erfani@unimelb.edu.au) and Shreyash (spatodia@student.unimelb.edu.au) via email (please use the subject header

“COMP30024 Project A Extension”) at the soonest possible opportunity (ideally **before** the deadline). If you have a medical reason for being late, you will be asked to provide a medical certificate. We will then assess whether an extension is appropriate. Requests for extensions on medical grounds received after the deadline may be declined. Note that computer systems are often heavily loaded near project deadlines, and unexpected network or system downtime can occur. You should plan ahead to avoid leaving things to the last minute, when unexpected problems may occur. Generally, system downtime or failure will not be considered as grounds for an extension.

Academic integrity

There should be one submission per group. You are encouraged to discuss ideas with your fellow students, but your program should be entirely the work of your group. It is not acceptable to share code between groups, nor to use the code of someone else. You should not show your code to another group, nor ask another group to look at their code. **If your program is found to be suspiciously similar to someone else's or a third party's software, or if your submission is found to be the work of a third party, you may be subject to investigation and, if necessary, formal disciplinary action.**

Please refer to the ‘Academic Integrity’ section of the LMS, and to the university’s academic honesty website http://academichonesty.unimelb.edu.au/ if you need any further clarification on these points.