

4511W, Fall-2021

WRITING ASSIGNMENT 1 :

**Assigned: 09/21/21 Due: 09/28/21 at 11:55 PM** (submit via Gradescope) Submit only a pdf

## Written

For this assignment you are asked run some tests and generate some data, then put this data into a table and figure/plot. Additionally, you have to analyze the results in a critical manner. All of the algorithms you will need to run are already implemented in the code (see below) (specifically, in search.py and it might also be useful to look at tests/test\_search.py). **I would not recommend trying to run test\_search.py, but rather just view the file for example usage.** Please make sure the computer you run this code on has **at least 8 gigs of RAM/memory** free. If you are using CSE Labs machines, all computers in Keller 1-250, 4-250 have more than enough.

## What questions to answer

(1) For the table, compare both the runtime and memory usage between depth limited search (depth limit=50 is fine) and breadth first search on the 8-puzzle using the initial state below. Write a paragraph describing the pros and cons between these two searches (BFS vs. depth limited search) on this problem. Also describe how applicable depth first search (not limited) is for this problem.

6	2	3
1	5	4
	7	8

(2) For the figure, gather data on depth first search and breadth first search for the N-queens problem. Specifically, gather runtime data (you do not need to plot the memory usage, though you can include in your discussion) for boards sized 8 to 13 for breadth first search and size 8 to 20 for depth first search. Plot this data in a figure (not a table) and discuss the pros and cons of using these algorithms for varying sizes of the N-queens problem.

## How to measure

Obviously, you will have to find the runtime and memory usage to answer the above questions. If you want to use a manual means (i.e. stopwatch and looking at the drop in memory at the end of the program (it will only go up) in the “System monitor” or your OS equivalent), that is fine. You can also get this info from python if you want (you might need to install some modules to use these commands). If you wish to take this approach, you are on your own... but here are some different ways to get this to work:

<https://stackoverflow.com/questions/11886862/calculating-computational-time-and-memory-for-a-code-in-python>

<http://www.marinamele.com/7-tips-to-time-python-scripts-and-control-memory-and-cpu-usage>

<https://www.geeksforgeeks.org/monitoring-memory-usage-of-a-running-python-program/>

If you are using a CSE labs machine, there are two commands you can use to get this info:

Runtime command: `time python3 filename.py`

Should give an output like:

```
0.006u 0.003s 0:01.57 0.0% 0+0k 56+0io 1pf+0w
```

You want just the first number (0.006) or “user” time.

Memory command: `grep VmPeak /proc/$PID/status`

... where the \$PID is the “process ID”, which you can find by typing “ps” while a program is running. Note, that you will have to run both the “ps” and “grep” with python still open, so you might want to add an extra line so the code waits for a keyboard command before quitting (obviously you can’t use this for the runtime info then).

So after you have python open in one window and a terminal in another (if SSHing you can just connect to the same machine twice) as my umnID is park0580, I would type “ps -u park0580” and see:

```
PID TTY      TIME CMD
1293 ?        00:00:00 systemd
1294 ?        00:00:00 (sd-pam)
1364 ?        00:00:00 sshd
1367 pts/2    00:00:00 tcsh
4318 pts/2    00:00:00 python3
4401 ?        00:00:00 sshd
4402 pts/0    00:00:00 tcsh
4882 pts/0    00:00:00 ps
```

Then I would type “`grep VmPeak /proc/4318/status`” as that is the python process to see:

```
VmPeak:      37112 kB
```

Note: there is a space on both sides of “VmPeak”.

## The code

The book provides code for the algorithms presented. For this class, we will use the python version of the code. Download the python code here:

<https://github.com/aimacode/aima-python>

I’d recommend **NOT** doing their install feature. You can simply download the code and start running it in python3 without doing anything else.

To use the code, I’d recommend making a file in the base directory (with things like “README.mb”). Then run python from that folder (such as with the command: `python3 myfile.py`). For this problem you should only need to use functions/classes from `search.py`, so just at the top of `myfile.py` have:

from search import \*

... then use the test\_search as a reference for how to use the functions. There is a Jupyter Notebook that has examples of how to use the algorithms (if that's your thing):

<https://github.com/aimacode/aima-python/blob/master/search.ipynb>

## LaTeX

The writing must be done in LaTeX and the resulting pdf submitted. ShareLatex/Overleaf is a website that allows you to write a "tex" file and view the resulting pdf in a web-browser. This eliminates the need to download Latex (though it is on the CSE labs machines), but requires you to register (personal accounts are free).

<https://www.sharelatex.com>

<https://www.overleaf.com/>

A sample LaTeX file is provided here (and resulting pdf) that demonstrates how to write text, make tables, and add pictures (i.e. the plot/figure):

<http://www-users.cselabs.umn.edu/classes/Fall-2021/csci4511-001/assignments/sample.zip>

<http://www-users.cselabs.umn.edu/classes/Fall-2021/csci4511-001/assignments/writing1sample.pdf>

For a tutorial on LaTeX, try google or here:

<http://www-users.cselabs.umn.edu/classes/Spring-2020/csci4511/misc/4511LatexGuide.html>

## Grading:

Latex 20%

Figure 20%

-Data presentation 10%

-Meets requirements 10%

Table 20%

-Data presentation 10%

-Meets requirements 10%

Analysis 40%

-Figure analysis content 10%

-Figure analysis clarity 10%

-Table analysis content 10%

-Table analysis clarity 10%