Dorian Benitez
drb160130

CS 4365.501
September 18, 2020

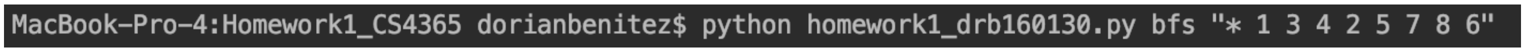
# Homework 1: README

## 1. How to run the program

Step 1: Download the “Homework1_drb160130.py" file
Step 2: Open a terminal/command prompt and re-direct to the file
Step 3: Type "python homework1_drb160130.py bfs "* 1 3 4 2 5 7 8 6"" to run the program and pass the search algorithm type and puzzle board design
Step 4 (Optional): Try different puzzle boards and algorithms. The program will read an input with each of the values [*,1,2,3,4,5,6,7,8] included, separated by a space

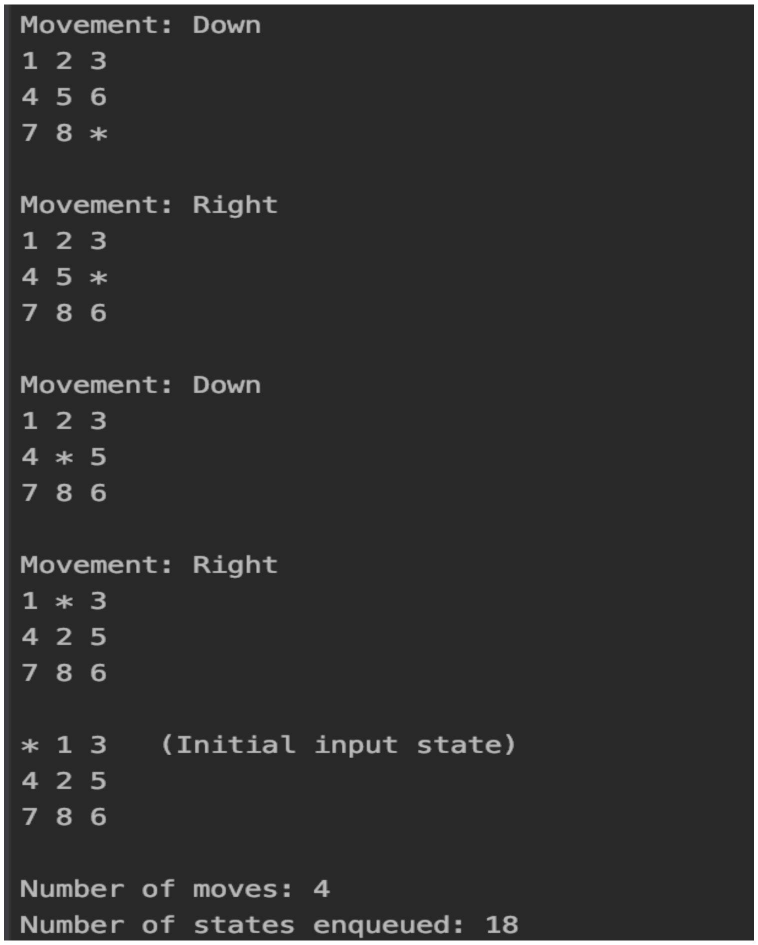
## 2. Sample input and corresponding output

*Input 1: After re-directing the terminal to the .py file, we perform a breadth-first-search*



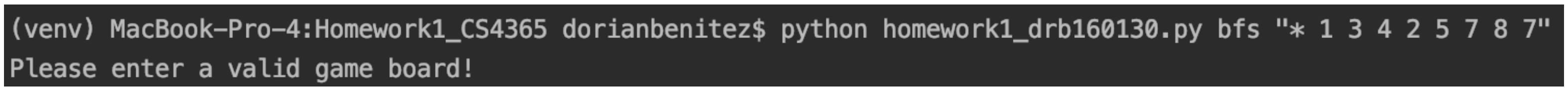
```
MacBook-Pro-4:Homework1_CS4365 dorianbenitez$ python homework1_drb160130.py bfs "* 1 3 4 2 5 7 8 6"
```

*Output 1: After running that line in the terminal, the following was the output*



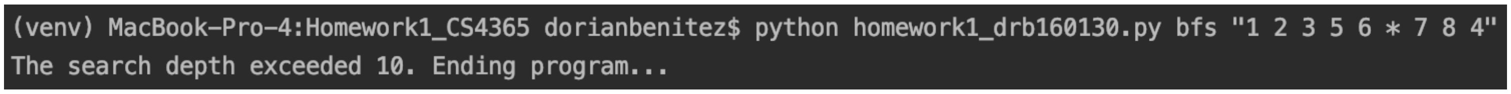
```
Movement: Down
1 2 3
4 5 6
7 8 *

Movement: Right
1 2 3
4 5 *
7 8 6

Movement: Down
1 2 3
4 * 5
7 8 6

Movement: Right
1 * 3
4 2 5
7 8 6

* 1 3   (Initial input state)
4 2 5
7 8 6

Number of moves: 4
Number of states enqueued: 18
```

*Input 2: We input an invalid puzzle board (has duplicate values)*
*Output 2: The user is prompted to try again with a valid puzzle input*



```
(venv) MacBook-Pro-4:Homework1_CS4365 dorianbenitez$ python homework1_drb160130.py bfs "* 1 3 4 2 5 7 8 7"
Please enter a valid game board!
```

*Input 3: We perform a search on a puzzle that exceeds a maximum search depth of 10*
*Output 3: The user is notified that the algorithm exceeded the maximum search depth*



```
(venv) MacBook-Pro-4:Homework1_CS4365 dorianbenitez$ python homework1_drb160130.py bfs "1 2 3 5 6 * 7 8 4"
The search depth exceeded 10. Ending program...
```

## **3. Comparative analysis of two heuristics used in A***

In the two A* search algorithms, we utilize two different heuristics. This is done because all search strategies are distinguished by the order in which nodes are expanded. Using two different heuristics, it allows us to obtain a more informed search for which non-goal state is more promising than another. The approach chosen to follow for this assignment was utilizing the best-first search. This is an instance of a general tree-search algorithm where a node is evaluated based on its function. Then, after obtaining the cost estimate for each node, we can evaluate which node would be best expanded first.

In this particular case, we can see that the first heuristic was a lot more accurate and beneficial than the second heuristic. This can be witnessed by observing the expanded nodes instantiated by each heuristic and the actual results of the algorithm. The first heuristic provides more information than the second and allows us to find the solution more efficiently.