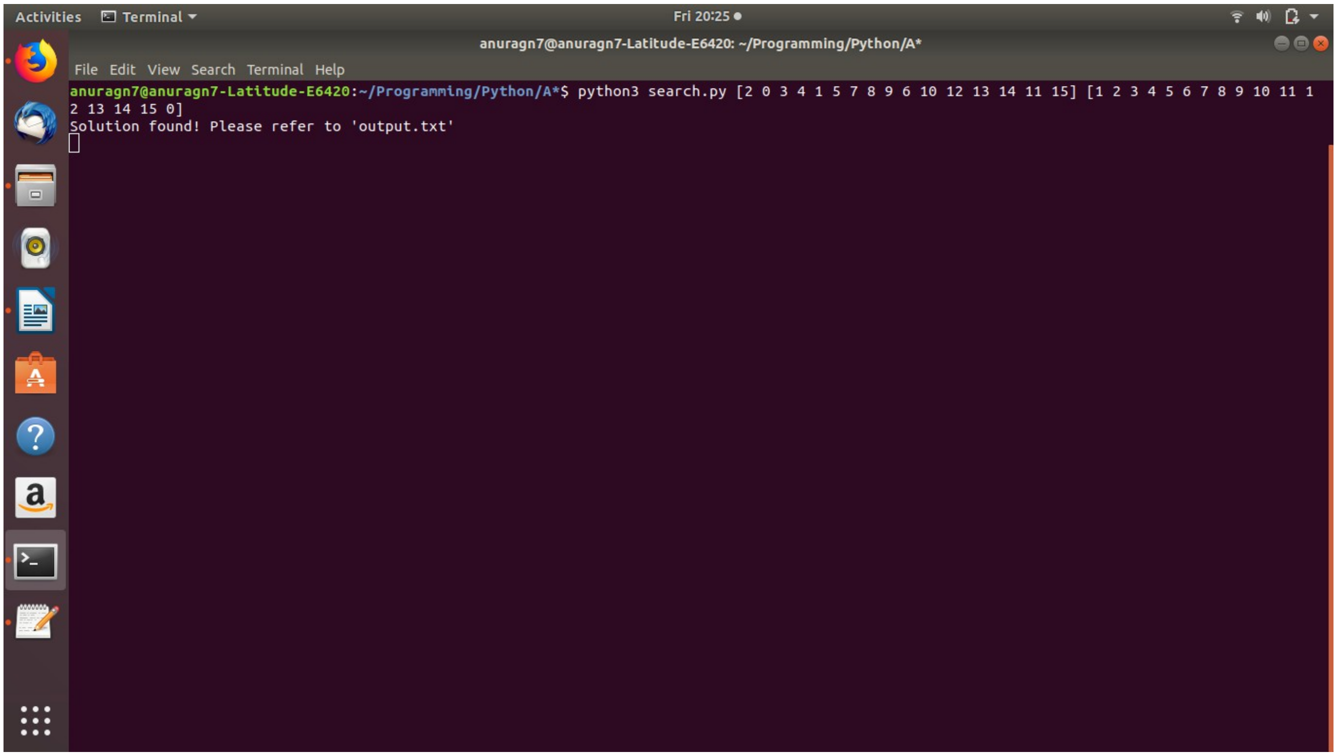
<u>Instructions:</u>

Once you're in the current directory you've unzipped this assignment to, in the command line type in: 'python3 search.py [start node] [goal node]'

Here is an example:



Activities
Terminal
Fri 20:25
anuragn7@anuragn7-Latitude-E6420: ~/Programming/Python/A*
File Edit View Search Terminal Help

```
anuragn7@anuragn7-Latitude-E6420:~/Programming/Python/A*$ python3 search.py [2 0 3 4 1 5 7 8 9 6 10 12 13 14 11 15] [1 2 3 4 5 6 7 8 9 10 11 1
2 13 14 15 0]
Solution found! Please refer to 'output.txt'
```

Once you've run your test case, you will see the program's output in the text file 'output.txt.'

<u>g(n) used:</u>

The step cost of moving from a node to it's successor remained the same (2), regardless of the tile being moved.

Rationale: The processing power and memory requirements needed to move a tile didn't change, irrespective of the actual tile being moved. Thus, it felt unnecessary to take the tile's value into account when formulating my g(n).

* I kept the step cost small to avoid running into the issue of the previous assignment, using the first first heuristic.

<u>h(n) used</u>:

Find the sum of the smallest number of moves each tile out of place needs to take to reach its destination. If the tile is only unit of distance away from its destination, it should not contribute to the overall score.

Rationale: I wanted the heuristic to focus on tiles that need to travel significant distances to reach their destination. Tiles one unit of distance away only needed a swap, thus I felt they could be ignored.

You can find the outputs for all three test cases in the file ‘results.txt.’