Task 6: Graphs, Search and Rules

# Towers of Hanoi: Random Search Results

Running the Towers of Hanoi random search code 10 times for Step 3 of the Tutorial, the program found successful patterns on:

* The 6th attempt: 32 attempted moves, 8 valid moves: [(0, 1), (0, 2), (2, 0), (1, 2), (2, 0), (0, 2), (0, 1), (2, 1), (0, 2), (1, 2), (2, 1), (2, 0), (0, 2), (1, 2), (1, 0), (2, 1), (0, 2), (1, 2)]
* The 8th attempt: 33 attempted moves, 12 valid moves: [(0, 1), (1, 2), (2, 0), (0, 2), (0, 1), (2, 1), (0, 2), (1, 2), (1, 0), (2, 1), (0, 2), (1, 2)]
* The 10th attempt: 55 attempted moves, 22 valid moves: [(0, 1), (0, 2), (2, 0), (1, 2), (2, 0), (0, 1), (1, 2), (2, 0), (0, 1), (1, 0), (0, 2), (0, 1), (2, 0), (0, 1), (1, 0), (0, 1), (1, 2), (2, 1), (0, 2), (1, 0), (1, 2), (0, 2)]

Doing a second run, the program found successful patterns on:

* The 3rd attempt: 55 attempted moves, 30 valid moves: [(0, 2), (2, 0), (0, 2), (2, 1), (1, 2), (2, 0), (0, 2), (0, 1), (2, 0), (0, 2), (2, 1), (0, 2), (1, 0), (0, 2), (2, 0), (0, 1), (2, 0), (1, 0), (0, 2), (2, 0), (0, 1), (0, 2), (1, 0), (0, 2), (2, 0), (1, 2), (2, 1), (1, 2), (0, 1), (1, 2)]
* The 7th attempt: 77 attempted moves, 38 valid moves: [(0, 1), (0, 2), (1, 2), (2, 0), (0, 2), (0, 1), (2, 1), (1, 2), (1, 0), (2, 1), (2, 0), (1, 2), (2, 1), (0, 2), (1, 0), (0, 2), (2, 0), (0, 1), (1, 0), (2, 1), (1, 2), (0, 2), (0, 1), (1, 0), (2, 1), (1, 2), (0, 1), (2, 0), (0, 2), (1, 0), (2, 0), (2, 1), (0, 1), (0, 2), (1, 0), (1, 2), (0, 1), (1, 2)]
* The 9th attempt: 38 attempted moves, 19 valid moves: [(0, 1), (0, 2), (2, 0), (1, 0), (0, 2), (0, 1), (2, 1), (0, 2), (1, 0), (0, 1), (1, 2), (1, 0), (2, 0), (0, 2), (0, 1), (2, 0), (1, 2), (0, 1), (1, 2)]

# Towers of Hanoi: Better Random Search

Running the Towers of Hanoi random search code that avoids state A 🡪 state B 🡪 state A moves for Step 4 of the Tutorial, the program found successful patterns on:

* The 1st attempt: 34 attempted moves, 15 valid moves: [(0, 1), (1, 2), (2, 0), (0, 1), (1, 2), (0, 1), (2, 0), (0, 1), (0, 2), (1, 2), (1, 0), (2, 1), (0, 2), (1, 0), (0, 2)]
* The 2nd attempt: 27 attempted moves, 12 valid moves: [(0, 1), (0, 2), (1, 0), (2, 1), (0, 1), (0, 2), (1, 2), (1, 0), (2, 1), (0, 2), (1, 0), (0, 2)]
* The 7th attempt: 30 attempted moves, 11 valid moves: [(0, 1), (1, 2), (0, 1), (2, 0), (0, 1), (0, 2), (1, 2), (1, 0), (2, 1), (0, 2), (1, 2)]
* The 10th attempt: 86 attempted moves, 39 valid moves: [(0, 1), (1, 2), (0, 1), (2, 0), (1, 2), (0, 2), (2, 1), (2, 0), (1, 2), (0, 1), (2, 0), (0, 1), (1, 2), (1, 0), (2, 1), (0, 2), (1, 2), (0, 1), (2, 1), (1, 0), (0, 2), (1, 0), (2, 0), (2, 1), (0, 2), (1, 0), (2, 1), (0, 2), (1, 2), (2, 0), (2, 1), (0, 1), (0, 2), (1, 2), (1, 0), (2, 1), (0, 2), (1, 0), (0, 2)]

On a second run, the program found successful patterns on:

* The 6th attempt: 34 attempted moves, 17 valid moves: [(0, 2), (2, 1), (1, 0), (0, 2), (0, 1), (2, 0), (0, 1), (0, 2), (1, 0), (0, 2), (2, 1), (1, 0), (0, 2), (2, 1), (1, 0), (1, 2), (0, 2)]
* The 7th attempt: 44 attempted moves, 21 valid moves: [(0, 1), (0, 2), (1, 2), (0, 1), (2, 0), (2, 1), (0, 2), (1, 0), (2, 0), (0, 1), (1, 2), (0, 1), (2, 0), (0, 1), (1, 2), (1, 0), (2, 0), (1, 2), (0, 1), (0, 2), (1, 2)]
* The 8th attempt: 51 attempted moves, 21 valid moves: [(0, 1), (0, 2), (1, 2), (2, 0), (0, 1), (2, 0), (1, 0), (0, 2), (2, 1), (0, 2), (1, 0), (2, 1), (0, 2), (2, 1), (0, 2), (1, 0), (0, 2), (2, 1), (1, 0), (1, 2), (0, 2)]
* The 9th attempt: 22 attempted moves, 7 valid moves (the optimal solution): [(0, 2), (0, 1), (2, 1), (0, 2), (1, 0), (1, 2), (0, 2)]
* The 10th attempt: 54 attempted moves, 22 valid moves: [(0, 1), (0, 2), (1, 2), (2, 0), (0, 1), (1, 2), (0, 1), (2, 0), (0, 1), (1, 2), (2, 0), (2, 1), (0, 2), (1, 0), (2, 0), (1, 2), (0, 1), (1, 2), (2, 0), (0, 1), (0, 2), (1, 2)]

Table : Comparison of the results of the pure random and non-reversing random searches

|  |  |  |
| --- | --- | --- |
|  | Random | Random and Non-Reversing |
| Success Rate | 6/20 | 9/20 |
| Average attempted moves | 42.4 | 48.3 |
| Average valid moves | 21.5 | 22 |

Compared to the pure random search, the non-reversing random search had a higher success rate for finding valid patterns, a marginally better average number of attempted moves, and almost the same average number of valid moves taken, making it overall the better algorithm for completing the Towers of Hanoi puzzle of the two.

# Towers of Hanoi Extension: Random Search with History

I added the code to get the random search to check the proposed state against a cache of previous states it has already been in. The program found successful patterns on:

* The 1st attempt: 16 attempted moves, 8 valid moves: [(0, 2), (0, 1), (2, 1), (0, 2), (1, 2), (2, 0), (1, 2), (0, 2)]
* The 6th attempt: 39 attempted moves, 12 valid moves: [(0, 1), (1, 0), (0, 2), (0, 1), (2, 0), (0, 1), (0, 2), (1, 2), (2, 0), (1, 2), (0, 1), (1, 2)]
* The 8th attempt: 72 attempted moves, 21 valid moves: [(0, 1), (1, 0), (0, 2), (0, 1), (2, 1), (1, 0), (1, 2), (0, 1), (1, 2), (0, 1), (2, 1), (1, 0), (2, 1), (0, 1), (1, 2), (1, 0), (2, 0), (1, 2), (0, 1), (0, 2), (1, 2)]

On a second run, the program found successful results on:

* The 2nd attempt: 91 attempted moves, 14 valid moves: [(0, 2), (2, 0), (0, 1), (0, 2), (1, 0), (2, 1), (0, 2), (2, 1), (0, 2), (1, 2), (2, 0), (1, 2), (0, 1), (1, 2)]
* The 3rd attempt: 19 attempted moves, 9 valid moves: [(0, 1), (0, 2), (1, 0), (2, 1), (0, 1), (0, 2), (1, 0), (1, 2), (0, 2)]
* The 5th attempt: 59 attempted moves, 21 valid moves: [(0, 1), (1, 2), (0, 1), (2, 0), (1, 2), (0, 1), (1, 2), (0, 1), (2, 0), (2, 1), (0, 1), (1, 2), (1, 0), (2, 1), (1, 0), (1, 2), (0, 2), (0, 1), (2, 0), (1, 2), (0, 2)]
* The 7th attempt: 66 attempted moves, 17 valid moves: [(0, 1), (1, 2), (0, 1), (2, 1), (1, 0), (1, 2), (0, 1), (1, 2), (0, 1), (2, 1), (2, 0), (1, 2), (2, 0), (1, 2), (0, 1), (0, 2), (1, 2)]
* The 10th attempt: 26 attempted moves, 11 valid moves: [(0, 1), (0, 2), (1, 0), (2, 1), (0, 1), (0, 2), (1, 2), (2, 0), (1, 2), (0, 1), (1, 2)]

|  |  |  |  |
| --- | --- | --- | --- |
|  | Random | Random, Non-Reversing | Random, Non-Repeating |
| Success Rate | 6/20 | 9/20 | 8/20 |
| Average attempted moves | 42.4 | 48.3 | 48.5 |
| Average valid moves | 21.5 | 22 | 14.1 |

Introducing the check against past states had minimal effect on its success rate or the average number of attempted moves when it did find a successful pattern, but substantially improved the average number of average valid moves required to complete the puzzle, cutting it by a third.

# Towers of Hanoi: Recursion

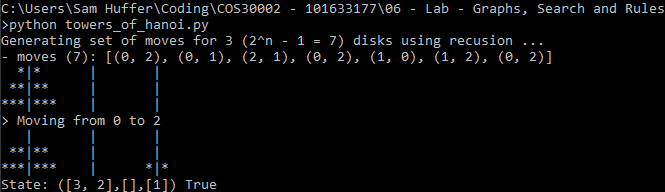
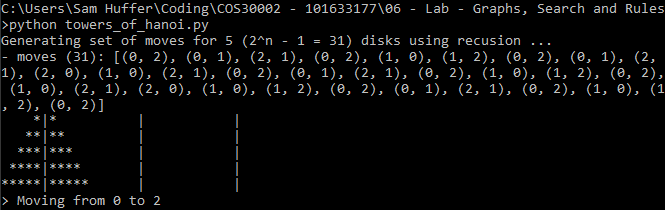
I completed the Towers of Hanoi extension using the code provided in the instructions and tested it against the default n = 3 scenario and an n =5 scenario.

Figure 2: Towers of Hanoi recursion with n = 5

Figure : Towers of Hanoi recursion with n = 3