Tower of Hanoi

27.01.2020

Lokesh Nirania 170010009

Sudarshan Das 170010010

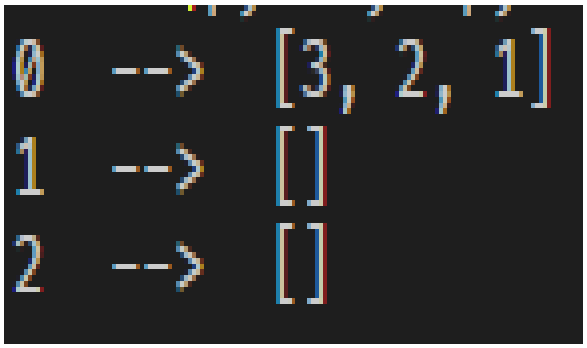
Brief Description about the Domain

Tower of Hanoi is a mathematical puzzle where we have three rods and n disks. The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:

1. Only one disk can be moved at a time.
2. Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack i.e. a disk can only be moved if it is the uppermost disk on a stack.
3. No disk may be placed on top of a smaller disk.

START STATE

An animated solution of the **Tower of Hanoi** puzzle for *T*(4, 3)



GOAL STATE

An animated solution of the **Tower of Hanoi** puzzle for *T*(4, 3)

```
0  -->  []
1  -->  []
2  -->  [3, 2, 1]
```

MoveGen(State)

// Try all combinations to move from disk i to disk j (i,j = 1 to 3)

// Accept only that state which follows the rule i.e., no bigger disk on top

For i = 1 to 3

    For j = 1 to 3

        If i != j // no point in swapping disks from same pole

            Swap from pole i to j

            If state valid then accept

            Else reject

GoalTest(State)

If State == goal

    Return true

Return false

Heuristic Function

Heuristic 1

```
h1_offset = 3 * ((discs)*(discs + 1)) / 2
// Last pole have highest weight whereas first pole have least weight
def heuristic_1(node):
    value = 0
        for i in range(0,3):
            for j in range(0,len(node.poles[i])):
                value = value + ((i + 1) *
                    node.poles[i][j])
    return h1_offset - value
```

// This way the goal state will have zero heuristic and start state have highest heuristic value

Heuristic 3

This Heuristic takes the weight of the poles, discs and its position on the pole and takes the sum of all the discs

For odd number of discs, all the odd numbered discs goes to the final pole whereas for even number of discs, all the even numbered discs goes to the final pole

The remaining numbered discs in both the cases goes to the middle pole

For the starting cases where maximum of the discs are on the starting pole, the weight is subtracted as it should be towards bottom of the hill

Best First Search

Discs = 3

Function	States Explored	Path Length
No Heuristic	25	9
Heuristic 1	9	8
Heuristic 2	27	16
Heuristic 3	8	8

Discs = 7

Function	States Explored	Path Length
No Heuristic	2145	129
Heuristic 1	680	172
Heuristic 2	2187	308
Heuristic 3	174	128

1. No Heuristic means BFS

   It gives the shortest path but states explored are very high

2. Heuristic 1

   It reduces the number of states explored but increases path length

3. Heuristic 3

   It's a very good heuristic with accuracy ~ 100%

Hill Climbing and Best First search

Disc = 7 and heuristic 3

	BFS	Hill Climb
States Explored	174	9
Time Taken	1.17 secs	0.00103 secs
Reaching the Optimal Solution	Yes	No

Tabu Search

Discs	Optimal Tenure	Path Length
7	40	237
5	8	42
4	3	16
3	2	8

Beam search analysis for different beam lengths

Discs = 3

Beam Width	States	Goal Reached
2	5	NO
3	5	NO
5	5	NO