**CSCE 625**

**Programming Assignment #1**

**Instructions for compilation:**

Python 2.7.6

$ python blocksworld.py arg1 arg2

Where arg1 is # of blocks and arg2 is # of stacks

The output generated is saved to a file called message.log

**My heuristic:**

I progressed on stepwise development of algorithm closely following this site: <http://www.cs.huji.ac.il/~ai/projects/2012/SolvingBlocksWorldProblems/>

1. Implemented # of out of place blocks algorithm
2. This heuristic is similar to Heuristic 1. It calculates the difference between the current state and the goal state, but looks at the details of each block. If Block A in the goal state is supposed to be on top of Block B and under Block C and in the current state it is neither on top of B or under C, then we add 2 to the heuristic.
3. This heuristic adds 2 for every block that is not currently directly on top of the block on which it has to be in the goal state or if there is such a block somewhere below it (in the same pile).
4. This heuristic is twice the number of blocks that must be moved once plus four times the number of blocks that must be moved twice. A block that must be moved once is a block that is currently on a block different to the block upon which it rests in the goal state or a block that has such a block somewhere below it in the same pile. A block that must be moved twice is a block that is currently on the block upon which it must be placed in the goal state, but that block is a block that must be moved or if there exists a block that must be moved twice somewhere below it (in the same pile).

**Queue generation:**

1. Pop out the block at the top of every stack and generate combinations by placing it on the other stacks
2. Introduced a check on the first stack of every state, while generating children nodes, to not disturb the perfect ascending order of the first stack and hence not to generate children by popping from the first stack.

**Observations:**

For the following initial state, the respective algorithms have yielded these results:

Initial State:

1 | B

2 | C E

3 | A D

* BFS:

success! depth= 10 , total\_goal\_tests= 2337 , max\_queue\_size= 2504

* H\_manhattan

success! depth= 10 , total\_goal\_tests= 577 , max\_queue\_size= 445

* H\_1

success! depth= 10 , total\_goal\_tests= 465 , max\_queue\_size= 370

* H\_2

success! depth= 10 , total\_goal\_tests= 183 , max\_queue\_size= 165

* H\_3

success! depth= 10 , total\_goal\_tests= 384 , max\_queue\_size= 299

* H\_4

success! depth= 11 , total\_goal\_tests= 145 , max\_queue\_size= 129

**Table of performance metrics:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # of blocks | # of stacks | depth | total\_goal\_states | max\_queue\_size |
| 5 | 3 | 14 | 302 | 281 |
| 5 | 3 | 7 | 25 | 32 |
| 5 | 3 | 11 | 51 | 50 |
| 5 | 3 | 10 | 112 | 123 |
| 5 | 3 | 8 | 58 | 72 |
| 5 | 3 | 9 | 85 | 96 |
| 5 | 3 | 7 | 17 | 17 |
| 5 | 3 | 12 | 366 | 335 |
| 5 | 3 | 10 | 141 | 166 |
| 5 | 3 | 7 | 84 | 95 |
| 6 | 3 | 12 | 52 | 53 |
| 6 | 3 | 14 | 712 | 761 |
| 6 | 3 | 12 | 93 | 111 |
| 6 | 3 | 11 | 169 | 236 |
| 6 | 3 | 15 | 301 | 373 |
| 6 | 3 | 13 | 511 | 628 |
| 6 | 3 | 12 | 301 | 350 |
| 6 | 3 | 15 | 299 | 399 |
| 6 | 3 | 7 | 35 | 41 |
| 6 | 3 | 12 | 345 | 404 |
| 7 | 3 | 16 | 629 | 766 |
| 7 | 3 | 17 | 390 | 529 |
| 7 | 3 | 19 | 2774 | 2976 |
| 7 | 3 | 17 | 743 | 1010 |
| 7 | 3 | 16 | 614 | 885 |
| 7 | 3 | 15 | 388 | 502 |
| 7 | 3 | 13 | 46 | 58 |
| 7 | 3 | 15 | 2532 | 2892 |
| 7 | 3 | 12 | 52 | 91 |
| 7 | 3 | 18 | 1563 | 197 |
| 8 | 3 | 16 | 124 | 189 |
| 8 | 3 | 19 | 2037 | 2951 |
| 8 | 3 | 18 | 2974 | 4238 |
| 8 | 3 | 15 | 118 | 203 |
| 8 | 3 | 20 | 6847 | 8618 |
| 8 | 3 | 15 | 346 | 463 |
| 8 | 3 | 17 | 1550 | 2150 |
| 8 | 3 | 23 | 5612 | 7233 |
| 8 | 3 | 16 | 1660 | 2152 |
| 8 | 3 | 23 | 4150 | 5150 |
| 5 | 4 | 10 | 118 | 197 |
| 5 | 4 | 11 | 472 | 815 |
| 5 | 4 | 7 | 43 | 105 |
| 5 | 4 | 10 | 401 | 704 |
| 5 | 4 | 12 | 708 | 1269 |
| 5 | 4 | 10 | 81 | 234 |
| 5 | 4 | 4 | 5 | 14 |
| 5 | 4 | 8 | 92 | 173 |
| 5 | 4 | 11 | 385 | 720 |
| 6 | 4 | 9 | 57 | 139 |
| 6 | 4 | 13 | 1250 | 2588 |
| 6 | 4 | 10 | 338 | 663 |
| 6 | 4 | 13 | 970 | 2082 |
| 6 | 4 | 7 | 63 | 144 |
| 6 | 4 | 8 | 59 | 155 |
| 6 | 4 | 7 | 34 | 94 |
| 6 | 4 | 13 | 1760 | 3748 |
| 6 | 4 | 10 | 78 | 204 |
| 6 | 4 | 7 | 65 | 172 |
| 7 | 4 | 14 | 454 | 1121 |
| 7 | 4 | 11 | 93 | 186 |
| 7 | 4 | 17 | 3535 | 8417 |
| 7 | 4 | 16 | 3046 | 6409 |
| 7 | 4 | 16 | 2514 | 5756 |
| 7 | 4 | 12 | 329 | 746 |
| 7 | 4 | 17 | 3326 | 9049 |
| 7 | 4 | 15 | 1590 | 4117 |
| 7 | 4 | 15 | 2699 | 6799 |
| 7 | 4 | 18 | 8323 | 19610 |
| 9 | 3 | 21 | 4860 | 6059 |
| 10 | 4 | 19 | 1751 | 4772 |

**Example program traces:**

$ python blocksworld.py 5 3

initial state:

1 | B

2 | C E

3 | A D

it= 0 , queue= 0 , f=g+h= 11 , depth= 0

it= 1 , queue= 5 , f=g+h= 10 , depth= 1

it= 2 , queue= 8 , f=g+h= 11 , depth= 2

it= 3 , queue= 9 , f=g+h= 11 , depth= 2

it= 4 , queue= 10 , f=g+h= 8 , depth= 3

it= 5 , queue= 9 , f=g+h= 11 , depth= 2

...

success! depth= 11 , total\_goal\_tests= 145 , max\_queue\_size= 129

solution path:

1 | B

2 | C E

3 | A D

1 | B

2 | C E D

3 | A

1 | B

2 | C E D A

3 |

1 |

2 | C E D A

3 | B

1 | A

2 | C E D

3 | B

1 | A B

2 | C E D

3 |

1 | A B

2 | C E

3 | D

1 | A B

2 | C

3 | D E

1 | A B C

2 |

3 | D E

1 | A B C

2 | E

3 | D

1 | A B C D

2 | E

3 |

1 | A B C D E

2 |

3 |

**Note:**

Due to insufficient time I have not been able to make the heuristic more efficient. If allowed, I would like to re-submit it at a later time.

**Bibilography:**

<http://www.cs.huji.ac.il/~ai/projects/2012/SolvingBlocksWorldProblems/>

<https://docs.python.org/2.7/>