**Heuristic:**

The heuristic function searches each stack from the bottom up to find the first incorrect block. It then counts all of the blocks on top of the incorrect block + 1, since all these will need to be moved and the block in the incorrect place will also need to be moved. This is done for each stack and the numbers are added up. This heuristic should be admissible, since in the best case scenario every block on top of the incorrectly placed block must be moved and the block must be moved to the right place.

**Examples:**

Current State:

['C', 'E']

['A', 'D']

['B']

Goal State:

[]

['A', 'D', 'B', 'C']

['E']

In this example, the first stack gets a value of 2. This is because the current state has 2 blocks in the first stack and the goal state has none, so they must be moved. The second column has a value of 0 because there are no blocks in the second stack that need to be moved. The value of the third stack is 1 since the block is in the wrong place and needs to be moved. The heuristic is the sum of these values, which is 3.

Current State:

['J']

[]

['B', 'C', 'E']

['A', 'D', 'G', 'I']

['F', 'H']

Goal State:

['J', 'I', 'G', 'C']

[]

['B', 'D', 'A']

[]

['F', 'H', 'E']

First stack: 0, there are no blocks in the current stack that are in the incorrect place

Second stack: 0, There are no blocks in the stack to move

Third stack: 2, the C is in the wrong place, so the C and E on top of it must be moved

Fourth stack: 4, all the blocks are in the wrong place

Fifth stack: 0, all the blocks in the current stack are in the right place

Overall: 6

**Summary:**

statistics: probA03.bwp planlen: 3 iters: 4 maxq: 10

statistics: probA04.bwp planlen: 4 iters: 9 maxq: 19

statistics: probA05.bwp planlen: 5 iters: 13 maxq: 31

statistics: probA06.bwp planlen: 6 iters: 27 maxq: 48

statistics: probA07.bwp planlen: 7 iters: 28 maxq: 55

statistics: probA08.bwp planlen: 8 iters: 57 maxq: 91

statistics: probA09.bwp planlen: 9 iters: 176 maxq: 198

statistics: probA10.bwp planlen: 10 iters: 352 maxq: 350

statistics: probA11.bwp planlen: 11 iters: 411 maxq: 477

statistics: probB03.bwp planlen: 3 iters: 6 maxq: 58

statistics: probB04.bwp planlen: 4 iters: 6 maxq: 75

statistics: probB05.bwp planlen: 5 iters: 7 maxq: 82

statistics: probB06.bwp planlen: 6 iters: 9 maxq: 83

statistics: probB07.bwp planlen: 7 iters: 88 maxq: 775

statistics: probB08.bwp planlen: 8 iters: 119 maxq: 903

statistics: probB09.bwp planlen: 8 iters: 106 maxq: 856

statistics: probB10.bwp planlen: 9 iters: 120 maxq: 973

statistics: probB11.bwp planlen: 9 iters: 125 maxq: 1049

statistics: probB12.bwp planlen: 9 iters: 41 maxq: 462

statistics: probB13.bwp planlen: 13 iters: 10797 maxq: 63189

statistics: probB14.bwp planlen: 13 iters: 8849 maxq: 54561

statistics: probB15.bwp planlen: 14 iters: 26150 maxq: 146825

statistics: probB16.bwp planlen: 15 iters: 20899 maxq: 114026

statistics: probB17.bwp planlen: 16 iters: 462581 maxq: 2010793

statistics: probB18.bwp planlen: 12 iters: 288 maxq: 2121

statistics: probB19.bwp planlen: 14 iters: 4880 maxq: 27299

statistics: probB20.bwp planlen: 15 iters: 65399 maxq: 325174