**Heuristic Functions**

H0 – returns 0, f = 0 + tree depth. Just BFS

H1 – Compares entire stacks between the current and goal state and returns the number of stacks that are not equivalent.

**H2 – (Best)** Counts how many blocks are not in the correct position in relation to the goal state and adds 1 point for each unit of size difference in each stack.

H3 – Performs a measurement of the number of moves required for a block to get put in the goal position. This heuristic performs marginally better than H2 for simple problems but does much worse on most of the more complicated/longer problems. Thus, H2 is better overall and more consistent.

**Table 1** Statistics of each problem output using H2

|  |  |  |  |
| --- | --- | --- | --- |
| **Problem Number** | **Solution Tree Depth** | **Iterations** | **Max Queue Size** |
| probA03.bwp | 3 | 3 | **16** |
| probA04.bwp | 4 | 15 | 74 |
| probA05.bwp | 5 | 20 | 108 |
| probA06.bwp | 6 | 65 | 320 |
| probA07.bwp | 7 | 100 | 502 |
| probA08.bwp | 8 | 468 | 2290 |
| probA09.bwp | 9 | 7728 | 38622 |
| probA10.bwp | 10 | 10362 | 50002 |
| probA11.bwp | 11 | 12570 | 61672 |
| probB03.bwp | 3 | 6 | 96 |
| probB04.bwp | 4 | 6 | 112 |
| probB05.bwp | 5 | 8 | 148 |
| probB06.bwp | 6 | 14 | 224 |
| probB07.bwp | 7 | 339 | 5600 |
| probB08.bwp | 8 | 839 | 12892 |
| probB09.bwp | 8 | 827 | 12964 |
| probB10.bwp | 9 | 460 | 7432 |
| ProbB11.bwp | 9 | 901 | 14740 |
| probB12.bwp | 9 | 88 | 1580 |
| probB13.bwp | FAILURE | FAILURE | FAILURE |