Nicolas Neven Lab 1 Extra Credit Results

For n = 1
2^1 x 1! = 2 x 1 =
2 possible combinations
All 2 combinations found
Running BFS/DFS on combinatons...
BFS Results: [1, 0]
BFS Average: 0.5
DFS Results: [1, 0]
DFS Average: 0.5

For n = 2
2^2 x 2! = 4 x 2 =
8 possible combinations
All 8 combinations found
Running BFS/DFS on combinatons...
BFS Results: [4, 3, 3, 2, 2, 1, 1, 0]
BFS Average: 2.0
DFS Results: [4, 3, 5, 6, 2, 1, 1, 0]
DFS Average: 2.75

For n = 3
2^3 x 3! = 8 x 6 =
48 possible combinations
All 48 combinations found
Running BFS/DFS on combinatons...
BFS Results: [6, 6, 5, 5, 5, 5, 5, 4, 5, 4, 4, 4, 4, 4, 4, 4, 4, 3, 4, 4, 4, 3, 3, 3, 3, 4, 3, 3, 3, 4, 3, 3, 4, 4, 3, 2, 3, 4, 4, 2, 2, 2, 2, 2, 1, 1, 1, 0]
BFS Average: 3.4375
DFS Results: [6, 7, 5, 7, 6, 8, 13, 8, 19, 4, 12, 9, 18, 5, 14, 18, 20, 3, 22, 18, 14, 9, 13, 17, 21, 4, 21, 17, 15, 10, 15, 3, 22, 23, 19, 10, 11, 16, 12, 2, 16, 20, 2, 16, 1, 1, 1, 0]
DFS Average: 11.520833333333334

For n = 4
2^4 x 4! = 16 x 24 =
384 possible combinations
All 384 combinations found
Running BFS/DFS on combinatons...

BFS Results: [8, 8, 8, 7, 7, 7, 7, 7, 7, 7, 7, 7, 6, 7, 7, 6, 6, 7, 6, 7, 6, 6, 7, 6, 6, 7, 6, 6, 6, 6, 6, 6, 6, 6, 6, 7, 6, 7, 6, 6, 6, 6, 5, 6, 6, 6, 6, 6, 6, 6, 6, 5, 6, 5, 5, 6, 6, 6, 6, 5, 6, 6, 6, 6, 6, 5, 6, 6, 6, 5, 6, 6, 6, 6, 6, 5, 5, 6, 5, 6, 6, 6, 6, 6, 5, 6, 5, 6, 6, 6, 5, 5, 5, 5, 5, 6, 5, 5, 5, 5, 6, 5, 5, 5, 5, 6, 5, 6, 6, 5, 6, 6, 6, 5, 5, 5, 6, 5, 6, 6, 5, 5, 5, 5, 4, 5, 6, 6, 6, 5, 5, 5, 6, 6, 5, 6, 5, 5, 5, 6, 6, 5, 4, 5, 5, 4, 5, 5, 5, 5, 5, 6, 6, 5, 6, 5, 5, 4, 5, 5, 5, 5, 6, 5, 5, 4, 6, 5, 5, 6, 4, 4, 5, 5, 6, 5, 5, 6, 4, 4, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 4, 5, 6, 5, 5, 4, 6, 5, 5, 4, 4, 4, 4, 6, 4, 4, 5, 6, 4, 4, 4, 5, 6, 5, 4, 4, 6, 5, 4, 6, 5, 5, 6, 5, 4, 4, 5, 5, 5, 4, 4, 4, 4, 4, 4, 4, 5, 5, 5, 6, 4, 5, 4, 4, 3, 4, 5, 6, 5, 6, 5, 4, 5, 5, 5, 4, 4, 5, 4, 4, 4, 6, 5, 4, 4, 3, 4, 4, 3, 4, 4, 5, 4, 5, 5, 5, 4, 4, 4, 4, 4, 4, 5, 4, 3, 5, 4, 3, 3, 4, 4, 4, 4, 5, 3, 3, 3, 4, 4, 4, 5, 4, 4, 4, 5, 4, 4, 5, 5, 5, 4, 3, 5, 3, 5, 3, 3, 3, 3, 3, 4, 5, 3, 5, 3, 4, 5, 4, 4, 3, 4, 4, 3, 4, 4, 4, 5, 5, 3, 5, 3, 4, 3, 4, 3, 4, 4, 3, 4, 3, 4, 3, 2, 4, 4, 3, 4, 3, 4, 3, 4, 2, 3, 2, 3, 3, 2, 2, 2, 3, 3, 3, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 0]

BFS Average: 4.796875

DFS Results: [8, 9, 9, 7, 9, 40, 8, 10, 90, 8, 10, 106, 128, 10, 27, 69, 6, 65, 119, 124, 105, 114, 11, 26, 70, 7, 41, 20, 106, 136, 19, 11, 38, 68, 7, 41, 107, 75, 19, 136, 143, 67, 70, 5, 39, 120, 28, 89, 25, 52, 30, 127, 11, 64, 19, 58, 64, 18, 90, 108, 69, 73, 42, 106, 100, 121, 135, 142, 66, 71, 6, 24, 83, 27, 88, 78, 51, 29, 126, 12, 21, 58, 106, 121, 17, 27, 5, 94, 42, 113, 74, 92, 18, 84, 71, 6, 40, 119, 100, 32, 24, 53, 128, 12, 91, 120, 123, 100, 25, 86, 126, 90, 18, 107, 46, 37, 67, 137, 38, 23, 78, 136, 53, 62, 12, 13, 118, 125, 104, 54, 68, 134, 105, 137, 37, 30, 26, 42, 25, 95, 39, 38, 4, 20, 57, 16, 133, 112, 59, 17, 83, 72, 134, 112, 105, 99, 31, 141, 43, 76, 99, 26, 87, 97, 88, 75, 136, 22, 57, 135, 61, 13, 138, 68, 31, 105, 133, 104, 36, 120, 43, 137, 63, 132, 107, 114, 107, 101, 90, 67, 118, 43, 23, 82, 28, 102, 14, 73, 122, 141, 34, 64, 45, 93, 39, 77, 111, 54, 102, 16, 143, 60, 86, 94, 85, 5, 46, 89, 109, 33, 72, 65, 97, 130, 28, 79, 50, 30, 121, 93, 113, 102, 134, 34, 118, 36, 96, 130, 117, 103, 111, 86, 135, 61, 66, 3, 4, 102, 83, 140, 101, 51, 24, 60, 133, 84, 82, 36, 129, 29, 92, 85, 139, 51, 54, 48, 13, 17, 108, 44, 137, 142, 142, 78, 110, 140, 141, 59, 110, 32, 66, 96, 29, 80, 31, 33, 95, 104, 60, 145, 4, 136, 139, 130, 110, 119, 35, 144, 21, 58, 84, 115, 111, 98, 32, 140, 45, 138, 88, 143, 23, 134, 112, 132, 3, 137, 80, 114, 59, 133, 63, 131, 116, 85, 80, 112, 132, 81, 50, 55, 47, 14, 137, 22, 21, 56, 62, 131, 47, 3, 63, 44, 144, 76, 103, 15, 47, 105, 135, 77, 86, 22, 55, 14, 135, 50, 15, 46, 136, 136, 49, 16, 2, 99, 61, 138, 49, 2, 56, 77, 143, 97, 35, 2, 145, 45, 98, 34, 48, 1, 1, 1, 1, 0]

DFS Average: 72.0546875

My computer could not finish computing all combinations for n = 5 and run BFS/DFS before the deadline. From these results, it seems to me that as the size of the problem gets larger (n increases), the BFS search algorithm finds shorter flip sequences on average. The larger the n, the larger the difference in performance between BFS and DFS. This output can be produced by executing extra_credit.py