Assignment 4 – AI intro

Group 256: Eirik Runde Barlaug & Jon Torgeir Grini

```
PS C:\Users\barla\AI-intro-ovinger\AI-intro-5> & C:/Users/barla/AppData/Local/Microsoft/Windo
ntro-5/O4/Assignment.py
Easy:
7 8 4 | 9 3 2 | 1 5 6
6 1 9 | 4 8 5 | 3 2 7
2 3 5 | 1 7 6 | 4 8 9
------+-------+------
5 7 8 | 2 6 1 | 9 3 4
3 4 1 | 8 9 7 | 5 6 2
9 2 6 | 5 4 3 | 8 7 1
------+-------+------
4 5 3 | 7 2 9 | 6 1 8
8 6 2 | 3 1 4 | 7 9 5
1 9 7 | 6 5 8 | 2 4 3
Solved with 0 failures, and 0 successes, with 1 calls to the backtrack() function
_____________________________________________________________________________________

Medium:
8 7 5 | 9 3 6 | 1 4 2
1 6 9 | 7 2 4 | 3 8 5
2 4 3 | 8 5 1 | 6 7 9
------+-------+------
4 5 2 | 6 9 7 | 8 3 1
9 8 6 | 4 1 3 | 2 5 7
7 3 1 | 5 8 2 | 9 6 4
------+-------+------
5 1 7 | 3 6 9 | 4 2 8
6 2 8 | 1 4 5 | 7 9 3
3 9 4 | 2 7 8 | 5 1 6
Solved with 0 failures, and 2 successes, with 3 calls to the backtrack() function
_____________________________________________________________________________________

Hard:
1 5 2 | 3 4 6 | 8 9 7
4 3 7 | 1 8 9 | 6 5 2
6 8 9 | 5 7 2 | 3 1 4
------+-------+------
8 2 1 | 6 3 7 | 9 4 5
5 4 3 | 8 9 1 | 7 2 6
9 7 6 | 4 2 5 | 1 8 3
------+-------+------
7 9 8 | 2 5 3 | 4 6 1
3 6 5 | 9 1 4 | 2 7 8
2 1 4 | 7 6 8 | 5 3 9
Solved with 4 failures, and 11 successes, with 12 calls to the backtrack() function
_____________________________________________________________________________________

Very Hard:
4 3 1 | 8 6 7 | 9 2 5
6 5 2 | 4 9 1 | 3 8 7
8 9 7 | 5 3 2 | 1 6 4
------+-------+------
3 8 4 | 9 7 6 | 5 1 2
5 1 9 | 2 8 4 | 7 3 6
2 7 6 | 3 1 5 | 8 4 9
------+-------+------
9 4 3 | 7 2 8 | 6 5 1
7 6 5 | 1 4 3 | 2 9 8
1 2 8 | 6 5 9 | 4 7 3
Solved with 57 failures, and 67 successes, with 68 calls to the backtrack() function
_____________________________________________________________________________________

PS C:\Users\barla\AI-intro-ovinger\AI-intro-5>
```

Reflection

It is clear that as the boards grows in difficulty, so does the number of calls to the backtrack function, as well as the failure count. For the easy board, the backtrack function is only called once, and the failure count is zero. This means that the AC3 algorithm managed to solve the board on its first try, and no recursive calls were needed. In the medium board, the backtrack function is called three times, but we still have a zero failure count. This means that for three of the variables, the algorithm had to pick out one arbitrary value in the variables domains, however the value was correct, and as a result the failure count did not increase. For the last two boards we have a non-zero failure count, meaning that for several of the instances of assignment our program did not find any matches i.e., the AC-3 algorithm wasn't able to shrink the domains of the variables enough to find any easy solution. The backtracking search must try different instances of assignments with different values for the variables many times for the last two boards to find a solution.