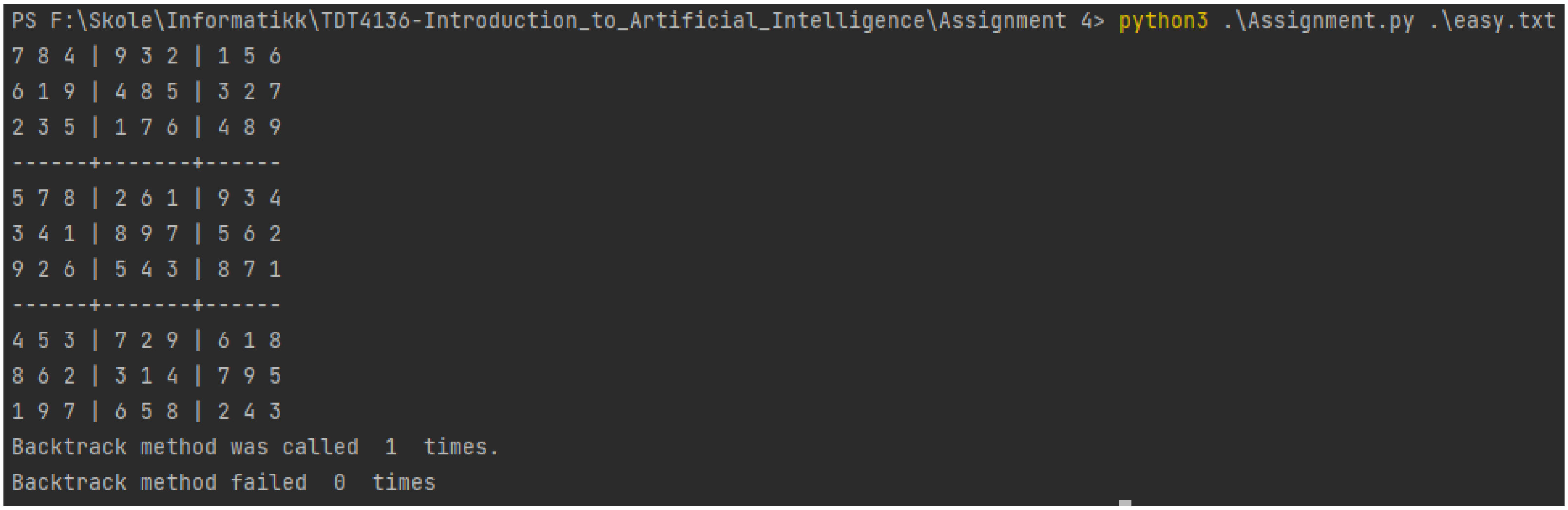
Assignment 4
TDT4136
Thomas Bakken Moe

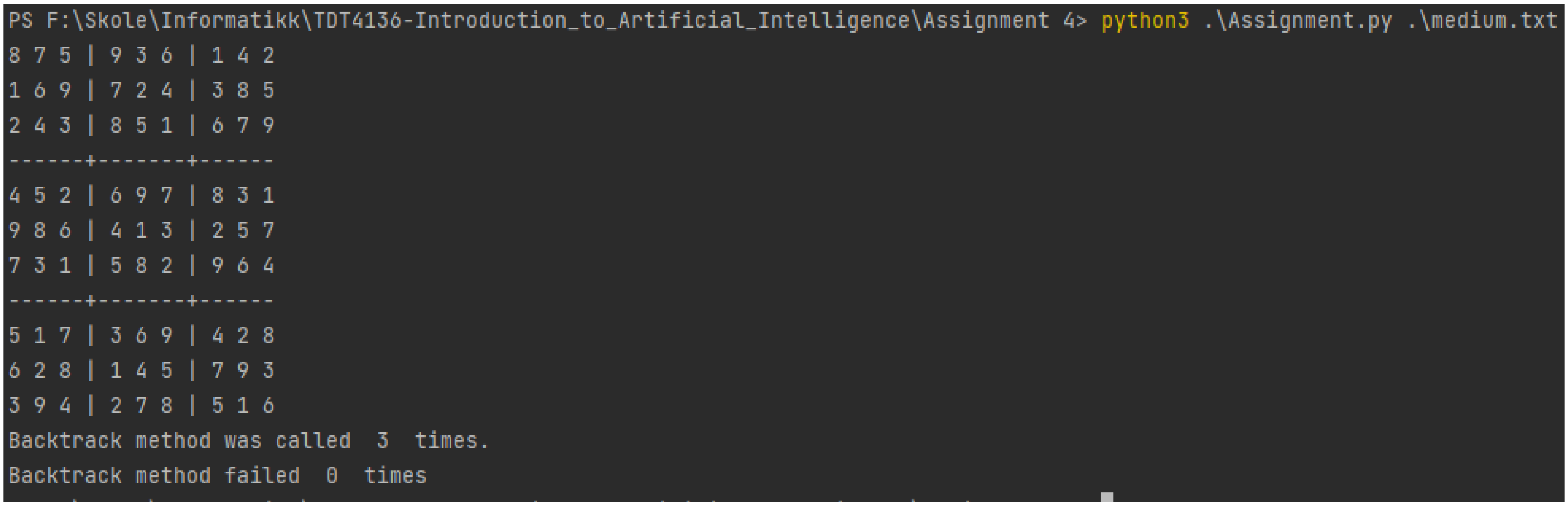
Board solutions

Easy:



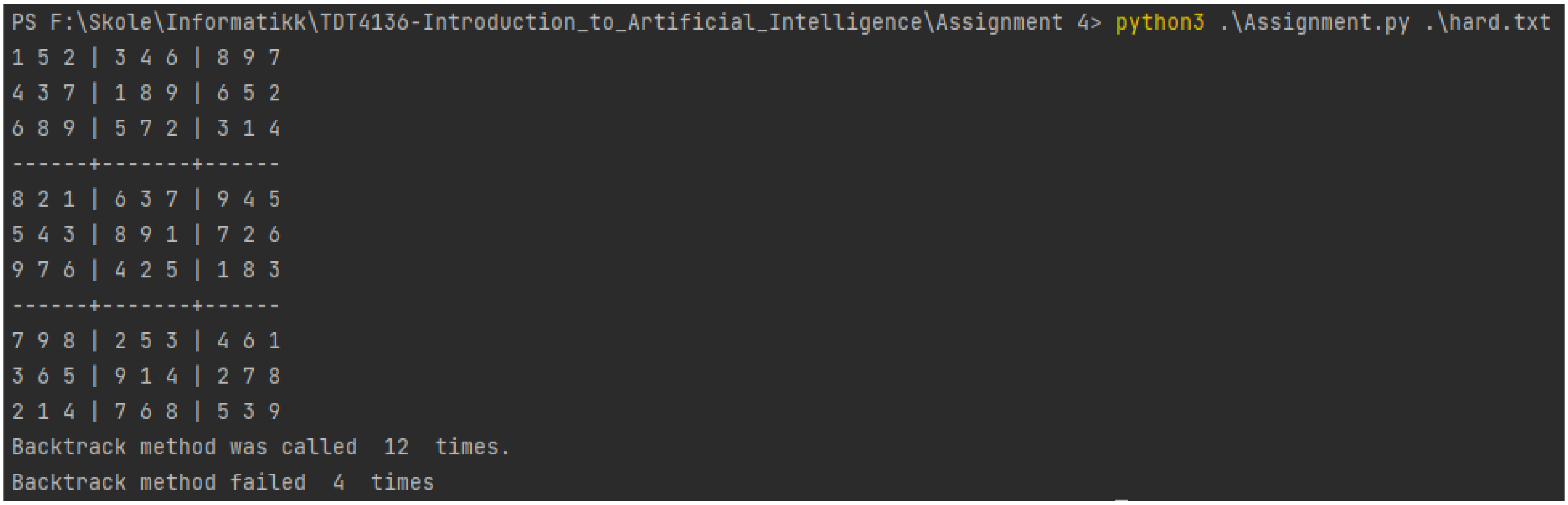
```
PS F:\Skole\Informatikk\TDT4136-Introduction_to_Artificial_Intelligence\Assignment 4> python3 .\Assignment.py .\easy.txt
7 8 4 | 9 3 2 | 1 5 6
6 1 9 | 4 8 5 | 3 2 7
2 3 5 | 1 7 6 | 4 8 9
------+-------+------
5 7 8 | 2 6 1 | 9 3 4
3 4 1 | 8 9 7 | 5 6 2
9 2 6 | 5 4 3 | 8 7 1
------+-------+------
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
Backtrack method was called  1  times.
Backtrack method failed  0  times
```

Medium:



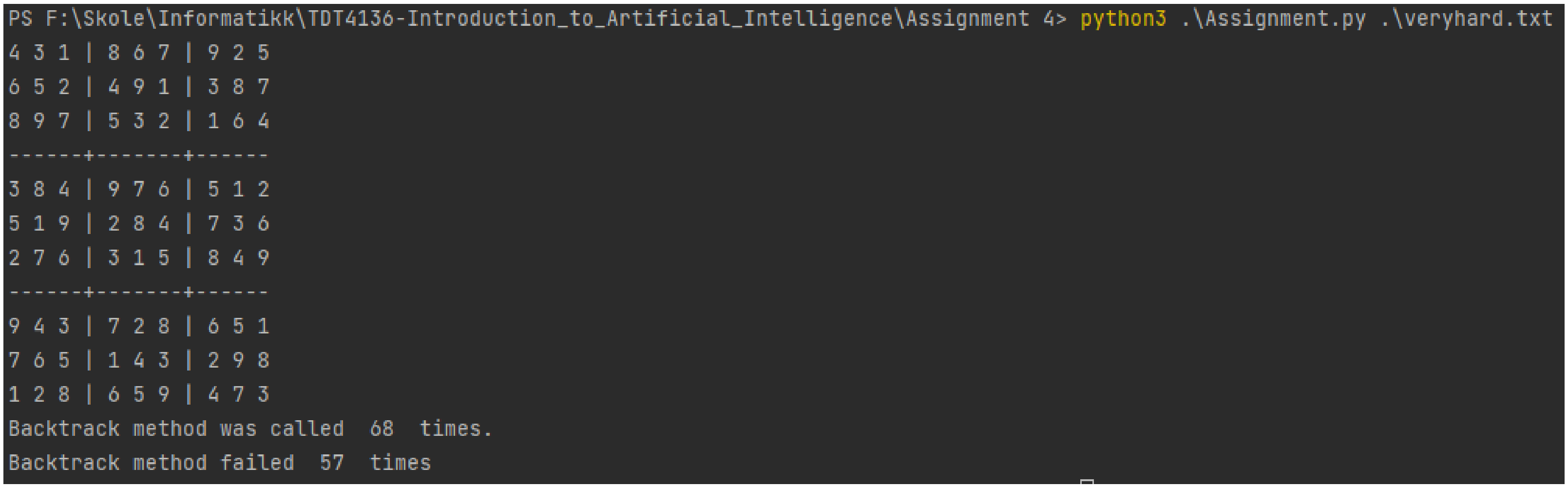
```
PS F:\Skole\Informatikk\TDT4136-Introduction_to_Artificial_Intelligence\Assignment 4> python3 .\Assignment.py .\medium.txt
8 7 5 | 9 3 6 | 1 4 2
1 6 9 | 7 2 4 | 3 8 5
2 4 3 | 8 5 1 | 6 7 9
------+-------+------
4 5 2 | 6 9 7 | 8 3 1
9 8 6 | 4 1 3 | 2 5 7
7 3 1 | 5 8 2 | 9 6 4
------+-------+------
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
Backtrack method was called  3  times.
Backtrack method failed  0  times
```

Hard:



```
PS F:\Skole\Informatikk\TDT4136-Introduction_to_Artificial_Intelligence\Assignment 4> python3 .\Assignment.py .\hard.txt
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
Backtrack method was called  12  times.
Backtrack method failed  4  times
```

Very hard:



```
PS F:\Skole\Informatikk\TDT4136-Introduction_to_Artificial_Intelligence\Assignment 4> python3 .\Assignment.py .\veryhard.txt
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
Backtrack method was called  68  times.
Backtrack method failed  57  times
```

Reflection

I was surprised by just how well the algorithm works for the easy and medium boards. Especially the easy board: the fact that backtrack() only has to be called once is incredible! I expected that it had to be called at least a few more times.

For the easier problems, we don't even have to use backtracking. The initial round of AC-3 is enough to solve the sudoku puzzle. This is very efficient and results in a really quick runtime.

However, when the initial AC-3 is not enough, we have to use backtracking. In this AC-3 + backtracking approach we are essentially splitting up into several “branches” based on the different legal values for every variable. For the harder problems, some of these “branches” will get stuck and fail. For the very hard puzzle we can see that the backtracking gets stuck almost every time. There might be an algorithm that is better suited to solve problems that are this difficult.