**Output board ”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 calls: 1

Backtrack failures: 0

”

Backtrack is only called once and never fails. This is because off the bat we have three cells with only one candidate. When these are filled in we have 8 new cells with only one candidate, which continuously gives new cells with one candidate. As there is no need to guess we have no fails.

**Output board ”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 calls: 3

Backtrack failures: 0

“

After some initial placement of cells with only one candidate, we arrive at a point where there are no more “givens”. Then the program runs through a cell with two candidates and checks if it can infer from the candidates whether a solution can be found. It does not fail with this inference, which is reflected in the backtrack failures statement.

**Output board ”Medium.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 calls: 12

Backtrack failures: 4“

Now we are getting to a hard-enough level to force a failure in the backtracking. It gets to a point where it must choose a value and check if it works for a recursive backtrack call with the new assignment. It does not and therefore have a backtracking failure.

**Output board ”Medium.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 calls: 68

Backtrack failures: 57

Here we have several calls and several failures. This is because we have no cells with have only one candidate off the bat. Solving a two-candidate cell does not lead to a one-candidate cell. This is the theme for this puzzle which is the reason to the high backtrack call and the high failure count.