

Raunak Chowdhury

I have three Sudoku solvers:

- 1) Naïve: The naïve search, nothing revolutionary. Sequentially fills in the board with possible values after generating a list of possible moves. Backtracks when no moves are found, replacing values from the list saved at that current state.
- 2) Smarter #1: A combination of two heuristics. The first is that it fills in all obvious solutions every iteration.
 - a) The second, which was generated from a helpful suggestion by Jonathan Singer (we did not work together however), is that given a certain position, you loop through all the cliques of that position and populate a list of 10 nested lists with the possible moves of that position. For instance, if position 6 yielded [2,3,7], the number “6” would be appended to the lists at indices 2,3, and 7. At the end, if any one nested list has a single value, then the position within that list must be the index of the list (if [6] was the only value in index 2, position 6 must be 2).
- 3) Smarter #2: Instead of filling values in in sequential order, the open positions with the least number of possible moves are prioritized first.

Results:

Note: r=ratio, b=backtracks, t=time elapsed

Board	Naïve	Smarter-1	Smarter-2
A1	b=260, t=0.002	b=0, t=0.001, r=inf	b=0, t=<0.001, r=inf
A2	b=27917, t=0.191	b=158, t=0.126, r=177	b=8, t=0.008, r=3490
A3	b=10616, t=0.075	b=43, t=0.036, r=247	b=93, t=0.08, r=114
A4	b=49498, t=0.456	b=334, t=0.237, r=148	b=0, t=0.006, r=inf
A5	b=39975, t=0.33	b=9, t=0.012, r=4444	b=0, t=0.01, r=inf
A6	b=6964, t=0.076	b=0, t=0.003, r=inf	b=0, t=0.004, r=inf
A7	b=2482069, t=25.553	b=3199, t=1.631, r=776	b=431, t=0.332, r=5759
A8	b=19029872, t=163.906	b=19115, t=12.068, r=996	b=1367, t=0.979, r=13921