CS171: Introduction to Artificial Intelligence
Professor Richard Lathrop
Sudoku Project Report

1. A. Generate a set of 5 boards with P = 3, Q = 3, M = 15.

   B. Run your Sudoku solver on the generated boards and fill in the following table:

	Backtracks	Trail Pushes
FC	179	4971
FC MRV	114	4321
FC LCV	829	12753
FC MRV LCV	1115	12665
NOR	70	3677
NOR MRV LCV	66	3666
NOR DEG LCV	516	10379
NOR MAD LCV	22	2430

   C. Which set of heuristics produced the best results? Explain.
   NOR MAD LCV had best results because it does the most constraint propagation, chooses the best possible variable, and the best possible value.

   D. Describe any anomalies or patterns. Explain in detail.
   Adding more heuristics resulted in better results but using NOR DEG LCV had the worst results. Choosing the variable with the most unassigned neighbors doesn't help very much, compared to choosing the variable with the smallest domain. Some boards also run really poorly for certain heuristics: Board4 – FC LCV, Board5 – FC MRV LCV

2. A. Generate a set of 5 boards with P = 4, Q = 3, M = 25.

   B. Run your Sudoku solver on the generated boards and fill in the following table:

	Backtracks	Trail Pushes
FC	6190	82172
FC MRV	233	9413

FC LCV	6336 + null	74715 + null
FC MRV LCV	48	6871
NOR	355	10268
NOR MRV LCV	126	7272
NOR DEG LCV	1603 + null + null	33288 + null + null
NOR MAD LCV	646	13759

C. Which set of heuristics produced the best results? Explain.
FC MRV LCV had the best results. Constraint propagation, choosing the variable with the smallest domain, and choosing the least constraining value worked the best. For a bigger board, it seems like it works better than using NOR and MAD.

D. Describe any anomalies or patterns. Explain in detail.
Using NOR and MAD resulted in worse results than when using FC or MRV. Using DEG still gave the worst results. It seems like bigger boards work better with FC and MRV. The tiebreaker in MAD and assignment in NOR don't affect the results positively. Also, LCV doesn't seem to be very useful without MRV.

3. A. Generate a set of 3 boards with P = 4, Q = 4, M = 45.

   B. Run your Sudoku solver on the generated boards and fill in the following table:

	Backtracks	Trail Pushes
FC	242 + null + null	6444 + null + null
FC MRV	860	20633
FC LCV	1037 + null	18499
FC MRV LCV	7	9168
NOR	19 + null + null	3144 + null + null
NOR MRV LCV	93	10683
NOR DEG LCV	Null, null, null	Null, null, null
NOR MAD LCV	61	9306

   C. Which set of heuristics produced the best results? Explain.

FC MRV LCV had the best results. Constraint propagation, choosing the variable with the smallest domain, and choosing the least constraining value worked the best. For a bigger board, it seems like it works better than using NOR and MAD.

D. Describe any anomalies or patterns. Explain in detail.
Using DEG resulted in runtimes that were too long. Using only NOR or only FC also resulted in super long runtimes. NOR and MAD resulted in worse results than when using FC or MRV. Also, LCV doesn't seem to be very useful without MRV. Having 3 heuristics yielded better results than only 1 or 2 heuristics.

4. Did the same set of heuristics produce the best results for (1), (2), and (3)? If not, explain why this might have happened.

   No. For boards bigger than p=3, q=3, it seemed like FC MRV LCV was better. However, for p=3, q=3 boards, NOR MAD LCV had the best results. The degree tiebreaker in MAD and value assignment in NOR don't seem to have a positive effect on the runtime.

5. Did you implement the Extra Credit heuristics? If so, which heuristics did you implement? Why did you choose these heuristics? How did they compare to the results in questions (1), (2), and (3)?

   N/A

6. Did you encounter any problem when doing the project? How did you resolve these problems?

   Yes. Sometimes we were confused about whether or not our code was correct or if our runtimes were accurate. We resolved these issues by going on Piazza and consulting with our classmates.

7. (Optional) Do you have any suggestions to improve the shells?