THE TEAM, AND HOW TO RUN THE CODE

The Team: TheWalkingZed

Me: Joseph Park, 24136956, josepjp2@uci.edu

Partner: None

By turning in this assignment, I do affirm that we did not copy any code, text, or data except CS-171 course material provided by the textbook, class website, or Teaching Staff.

Programming Languages and Versions used:

- C++, c++11

Environment Setup Required to Run:

- Compiled via MinGW 4.7.2 32-bit and Qt 5.5
- Delivered executable compiled on openlab with gcc 5.2.0, using commands:
  - –static-libstdc++
  - –static-libgcc
  - –std=c++11

Implementation:

- Variable & Domain – Variable is a combination of an array position and a domain. The Domain is a vector of possible character values.
- Puzzle – Holds the puzzle and its attributes as a 2-dimensional array of variables (array locations, combined with a character face value and a domain of character values)
- Generator – Generates a puzzle based on the attributes of m (number of cells pre-assigned), n (size of the puzzle), p (row size of blocks), and q (column size of blocks).
- Reader – Reads and writes puzzles from and to files.
- Solver – Solves a puzzle based on a variety of heuristic methods, each of which can be toggled on and off. It generates a log file after solving the puzzle.
- Book Keeper – Stores assignments, removals, and search depth after every assignment.
- Logger – Writes to an output file with various attributes from the solver.

WHAT MY TEAM DID

I coded it.			I tested it thoroughly.			It ran reliably and correctly.			What was it?
Yes	Partly	No	Yes	Partly	No	Yes	Partly	No	
Required Coding Project									
X			**X**			**X**			Backtracking Search (BT)
X			**X**				**X**		Forward Checking (FC)
X			**X**				**X**		Minimum Remaining Values (MRV)
X				**X**				**X**	Degree Heuristic (DH)
X				**X**				**X**	Least Constraining Value (LCV)
Extra Credit									
X			**X**			**X**			Writing My Own Shell
X			**X**			**X**			Writing My Own Random Problem Generator
X				**X**				**X**	Arc Consistency AC-3/ACP/MAC
		X			**X**			**X**	Local Search Using Min-Conflicts Heuristic
		X			**X**			**X**	Advanced Techniques, Extra Effort, or Creativity Not Reflected Above*

*Advanced Techniques, Extra Effort, or Creativity Not Reflected Above:

- None.

ANALYSIS OF BEST METHODS COMBINATION

FC	MRV	DH	LCV	(ACP)	(MAC)	(OTHER)	Average # Nodes	Average Time	Std. Dev. Time
The blank row below is for the case of no heuristics and no constraint propagation.									
							161576.7	1.1	2.385372
X							161576.7	2.6	5.986652
	X						161576.7	2.5	5.69649
		X					574099.6	Timeout	Timeout
			X				161576.7	9.8	20.89402
				(X)			161576.7	1.1	2.385372
					(X)		1276.1	3.0	4.898979
						(X)	N/A	N/A	N/A
X	**X**	**X**	**X**				927705.4	71.3	115.8594
X	**X**	**X**	**X**	**X**	**X**		1105.7	2.6	7.144228
Fill in the blank row above with the combination you found to be fastest on PH1-5.									

Note: These tests were run on 10 randomly generated 9x9 sudoku puzzles with M = 17, as stated in the professor's email., with timeout at 600 seconds.

Did you get the results you expected? Why or why not?

As a result of a less than stellar implementation of the heuristics and propagation, these operations end up taking more time to work than it reduces in the backtracking search. As you can see the fastest one is the one without any heuristics or propagation, or the one with only arc consistency preprocessor. While the degree heuristic and the least constraining value heuristic increases the average solution time by a large amount, combining them with the other heuristics and propagations causes the time to drop to around 2.6 seconds, with a drastically lower average node count, 1105.7 nodes. These were more or less the results I expected, although I expected the ACP + MAC to be much lower, and that all 6 combined would result in a much larger time.

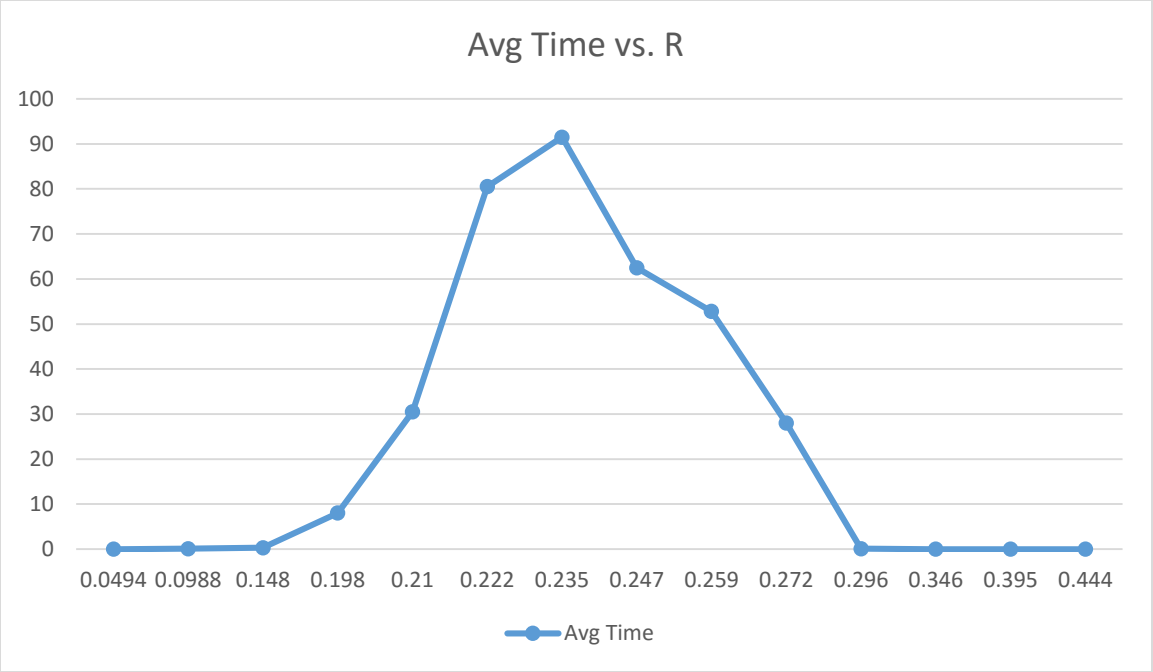
Did you implement any Advanced Techniques, Extra Effort, or Creativity not reflected above? If so, please tell us what you did.

No.

ESTIMATION OF THE CRITICAL VALUE, HARDEST R

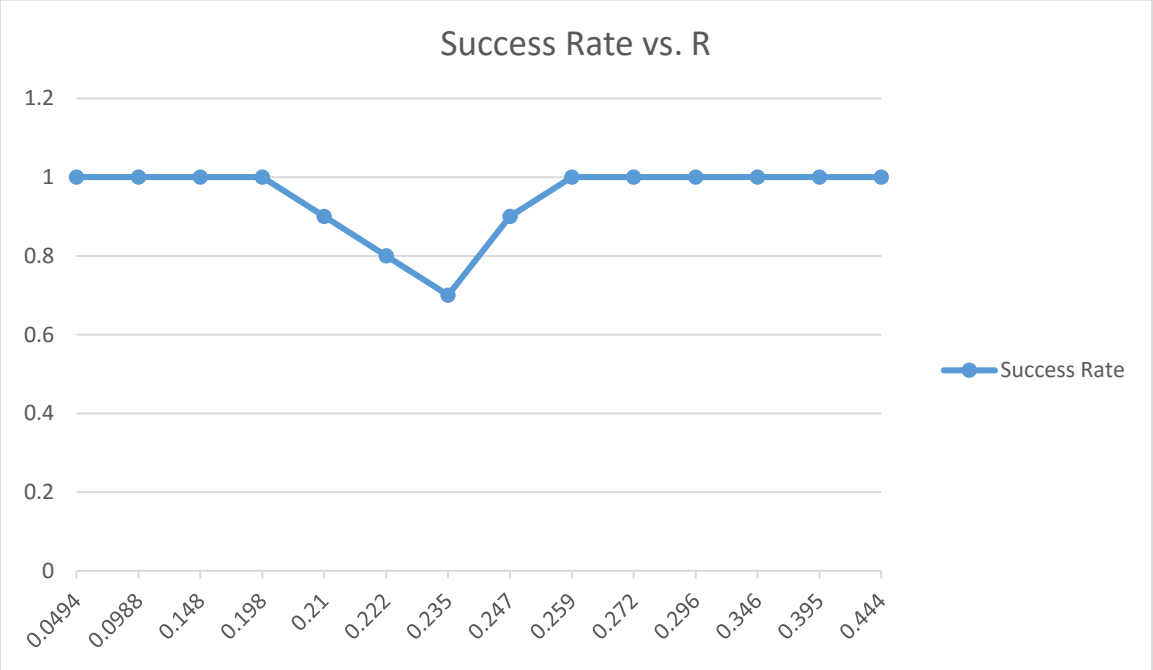
M	N	P	Q	R $= M/N^2$	Average # Nodes	Average Time	Std. Dev. Time	# (%) Solvable
4	**9**	**3**	**3**	**0.0494**	329.8	0.0	0.0	100%
8	**9**	**3**	**3**	**0.0988**	4095.9	0.1	0.3	100%
12	**9**	**3**	**3**	**0.148**	45535.3	0.3	0.9	100%
16	**9**	**3**	**3**	**0.198**	1714470.0	8.0	18.60645	100%
17	**9**	**3**	**3**	**0.210**	6747953.0	30.5	90.17455	90%
18	**9**	**3**	**3**	**0.222**	13713488.0	80.5	124.9066	80%
19	**9**	**3**	**3**	**0.235**	16755347	91.5	137.1869	70%
20	**9**	**3**	**3**	**0.247**	9505533	62.5	103.7663	90%
21	**9**	**3**	**3**	**0.259**	8360421	52.8	92.8502	100%
22	**9**	**3**	**3**	**0.272**	4896441	28	57.89646	100%
24	**9**	**3**	**3**	**0.296**	4687.7	0.1	0.3	100%
28	**9**	**3**	**3**	**0.346**	10628.6	0	0	100%
32	**9**	**3**	**3**	**0.395**	611.8	0	0	100%
36	**9**	**3**	**3**	**0.444**	184.9	0	0	100%

Find the critical value of the "hardest R" for $N = 9$ and your best combination above.



Based upon your results above, you estimate the "hardest R_9" = 0.235.

How does puzzle solvability for your best combination vary with $R = M/N^2$?



Is the critical value for "hardest R" approximately the same as the value of R for which a random puzzle is solvable with probability 0.5?

From the sample population of the tests I have conducted, the "hardest R" value estimated from the first graph coincides with the critical point in this graph, 0.7. It is not the same value of R for a probability of 0.5 (as there is none from my sample), but it is as close as it can get to it.

LARGEST N COMPLETABLE AT THE HARDEST R

"Hardest M" round ($N^2 \times R_9$)	N	P	Q	# (%) Completed in 5 Minutes or Less	AVERAGE # NODES (Completed puzzles only)	AVERAGE TIME (Completed puzzles only)	STD.DEV. TIME (Completed puzzles only)
34	**12**	**3**	**4**	70%	318225.6	2.428571	4.494895
53	**15**	**3**	**5**	40%	10606751	112	115.0326
60	**16**	**4**	**4**	10%	703576	9	None
76	**18**	**3**	**6**	20%	2940854	40.5	39.5
94	**20**	**4**	**5**	0%	0	0	None
104	**21**	**3**	**7**	20%	18148210	51.5	0.5
135	**24**	**4**	**6**	0%	0	0	None
171	**27**	**3**	**9**	0%	0	0	None
184	**28**	**4**	**7**	0%	0	0	None
212	**30**	**5**	**6**	0%	0	0	None
241	**32**	**4**	**8**	0%	0	0	None
288	**35**	**5**	**7**	0%	0	0	None

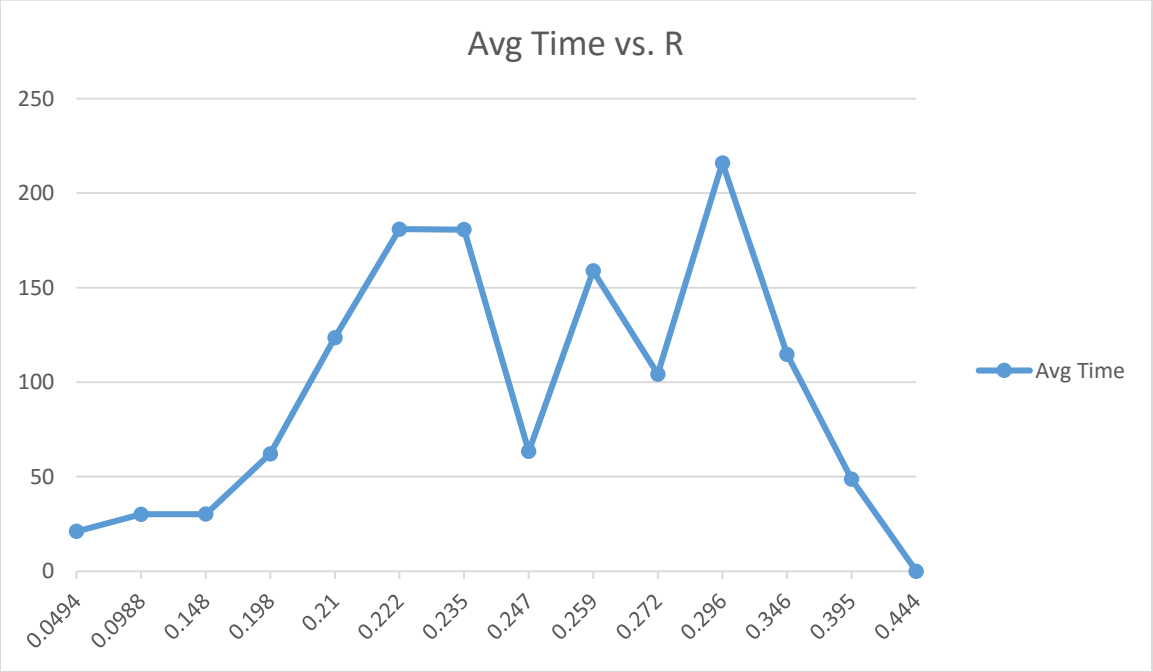
Based on the table above, the "Largest N" is 12, as it had the same probability of completion (70%) with the same R as 9.

IS HARDEST R CONSTANT AS THE SUDOKU PUZZLES SCALE?

M	N	P	Q	$R = M/N^2$	AVERAGE # NODES	AVERAGE TIME	STD.DEV. TIME	# (%) SOLVABLE
7	**12**	**3**	**4**	**0.0494**	2357400	21.1	58.44493	100%
14	**12**	**3**	**4**	**0.0988**	3403260	30.1	124.1055	90%
21	**12**	**3**	**4**	**0.148**	6403874	30.2	90.23425	90%
29	**12**	**3**	**4**	**0.198**	6578884	62.1	150.0642	80%
30	**12**	**3**	**4**	**0.210**	12843255	123.6	144.9581	60%

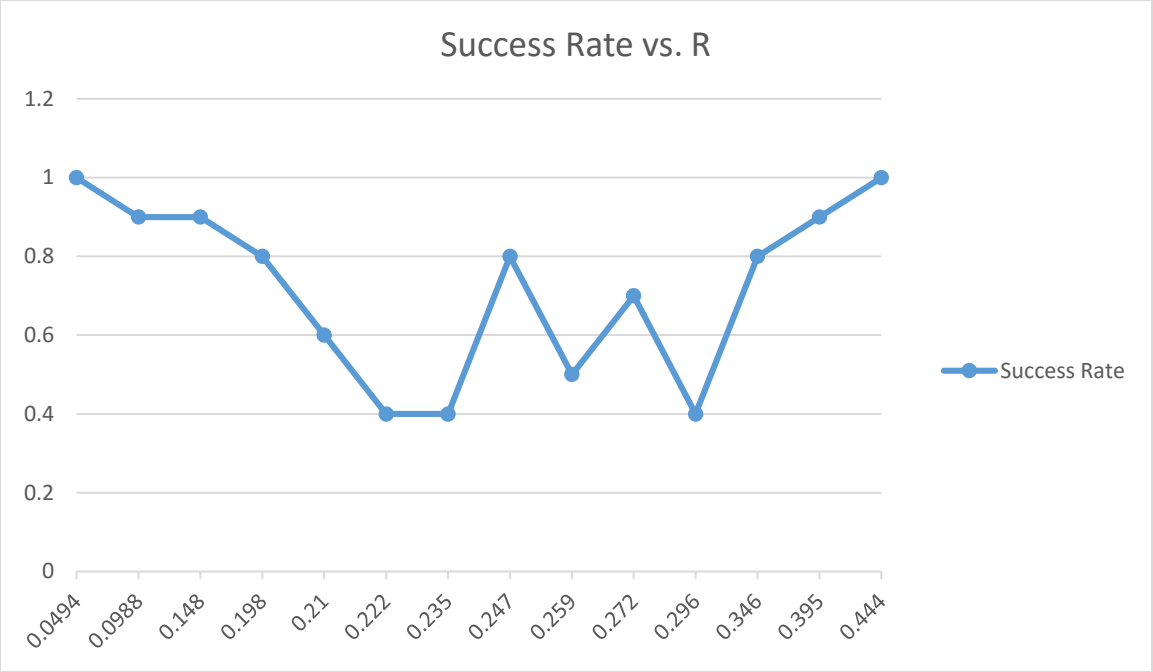
32	**12**	**3**	**4**	**0.222**	19227834	181	146.9714	40%
34	**12**	**3**	**4**	**0.235**	19788930	180.7	141.1602	40%
36	**12**	**3**	**4**	**0.247**	5790776	63.4	93.81258	80%
37	**12**	**3**	**4**	**0.259**	16523606	158.9	138.9141	50%
39	**12**	**3**	**4**	**0.272**	12112925	104.3	139.2179	70%
43	**12**	**3**	**4**	**0.296**	28076684	216	117.978	40%
50	**12**	**3**	**4**	**0.346**	11959403	114.8	138.8156	80%
57	**12**	**3**	**4**	**0.395**	5670697	48.7	53.48878	90%
64	**12**	**3**	**4**	**0.444**	3314.8	0	0	100%

Find the critical value of the "hardest R" for $N_{largest}$



Based upon your results above, you estimate the "hardest $R_{N_{largest}}$" = 0.296

How does puzzle solvability for your best combination vary with $R = M/N^2$?



Is the critical value for "hardest R" approximately the same as the value of R for which a random puzzle is solvable with probability 0.5?

The hardest R has around solvability probability of 0.4. The R values for which the solvability probability is around 0.5 would be around 0.259.

Is hardest $R_{N_{largest}} \approx$ "hardest R_9"? Why or why not? How is the hardest R related to the board parameter N? What is the shape of the function $R(N)$, which gives you the hardest ratio for $9, 12, 15, 16, 18, 20, \ldots, 35$?

The hardest $R_{N_{largest}} = 0.296$ is not approximately the hardest $R_9 = 0.235$. However, based on the tests, 0.296 and 0.235 had the same solvability probability at 0.4. 0.296 had a slightly higher average time, which is why this value is considered to be the hardest R for the largest N, $N = 12$. As the puzzles scale, the number of cells, assignments, and backtracks increases. Thus, larger puzzles will take longer to complete, leading to higher average times and more chances of timeout. Based on these tests, the hardest R increases as N increases although more tests at different Ns and hardest Rs are required for a more accurate relationship.