Hexadecimal Sudoku Solver

CS 271 | Fall 2017

Introduction to Artificial Intelligence

Shraavanth Penugonda	**35764082**
Srisruthi Sridhar	**49523982**
Adithya Srinivasan	**47491715**
Ramkumar Rajabaskaran	**85241493**

Problem Description:

- To adopt searching techniques into the given hexadecimal sudoku problem and find the correct solution efficiently.
- The puzzle consists of a 16X16 grid - divided into 16 4x4 sub-grids.
- The objective is to fill the grid with values from 0 to F such that each row, column and sub-grid contains all the values from 0 to F.

Problem Approach:

A sudoku problem can be solved using various algorithms separately or multiple combined. Some of the approaches to solve the problem are:

- Brute Force (Works only for lower order sudokus)
- Stochastic Search
- Backtracking Search
- Backtracking with forward checking
- Constraint Satisfaction
- Minimum Remaining Values(MRV)
- Least constraining value(LCV)
- Naked Twins Rule

The approach we followed is as follows:

- Modeled the puzzle as a Constraint Satisfaction Problem (CSP)
- Used Heuristics such as MRV and LCV to improve performance

- Incorporate Value based lookup (intuitive)
- Backtracking with forward checking

Constraint Network:

Variables	256 cells
Domain	[-1 to F]
Constraints	In each row, column, or sub-grid, the same digit cannot appear twice.

Value Based Lookup:

This is an intuitive solving technique that mimics the pen and paper method of finding a solution for sudoku. In this approach, we look from the value's perspective instead of looking at the domain of every cell. For e.g., Consider cell(A1), we look at the domain of the cell 1,2,3,4 etc. In value based lookup, we look at the value (1), and check if it is the only instance among its peer's domains (if 1 is present in only one of the peer's domains).

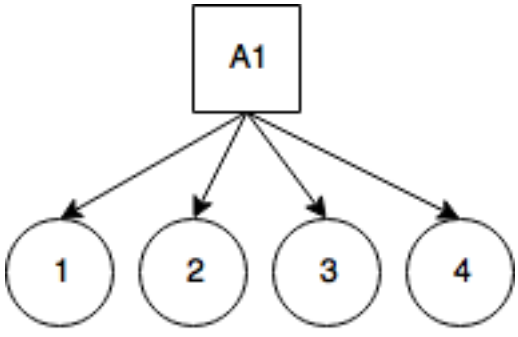


Fig A. Looking from Variable's Perspective

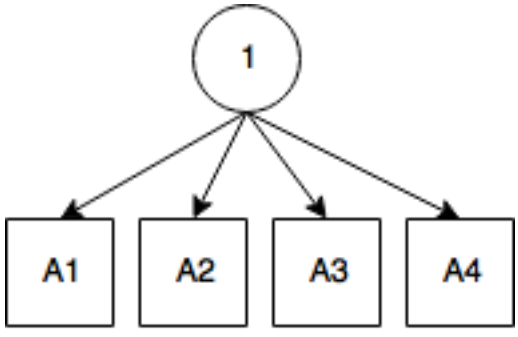
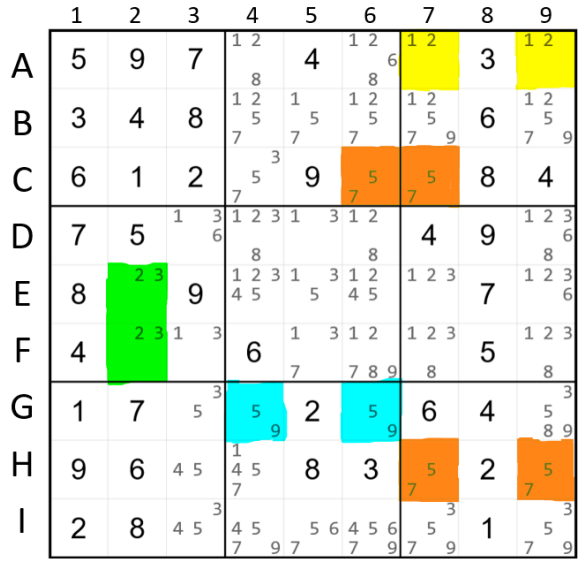


Fig B. Looking from Value's Perspective

Naked Twins Rule:

This is a technique that can be used to reduce the domains of certain cells if by eliminating values that have no possibility of occurring in that cell.



Two cells that belong to the same unit and have the same domain, of length 2 form a Naked Twin.

Consider Cells A7 and A9

They form a Naked Twin as they have the same domain {1,2}

Hence, we can be sure that 1,2 come in either A7 or A9 and hence we can remove 1,2 from the domains of its common peers. Similarly {C6, C7}, {E2, F2}, {G4, G6} form Naked twins

Representation:

The initial step was to represent the grid as follows:

1. The rows are named using alphabets [A-P].
2. The columns are named using hex digits[0-F].
3. 'Squares' is a list containing all the possible cells of the sudoku. (A0, A1, ..., PF).
4. For every Square, there is an entry in 'peers' with the cell name as the key and all possible neighbors of the cell as it's values.
5. List of domains of all the variables is maintained as a hash map(dict_domain).

6. The current state of the variables is inferred from 'dict_domain' which stores the possible values(domains) for every cell.
7. Peers are a set of squares that share a constraint

A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF
G0	G1	G2	G3	G4	G5	G6	G7	G8	G9	GA	GB	GC	GD	GE	GF
H0	H1	H2	H3	H4	H5	H6	H7	H8	H9	HA	HB	HC	HD	HE	HF
I0	I1	I2	I3	I4	I5	I6	I7	I8	I9	IA	IB	IC	ID	IE	IF
J0	J1	J2	J3	J4	J5	J6	J7	J8	J9	JA	JB	JC	JD	JE	JF
K0	K1	K2	K3	K4	K5	K6	K7	K8	K9	KA	KB	KC	KD	KE	KF
L0	L1	L2	L3	L4	L5	L6	L7	L8	L9	LA	LB	LC	LD	LE	LF
M0	M1	M2	M3	M4	M5	M6	M7	M8	M9	MA	MB	MC	MD	ME	MF
N0	N1	N2	N3	N4	N5	N6	N7	N8	N9	NA	NB	NC	ND	NE	NF
O0	O1	O2	O3	O4	O5	O6	O7	O8	O9	OA	OB	OC	OD	OE	OF
P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	PA	PB	PC	PD	PE	PF

Algorithm Description:

1. The given input state is assigned to the cells of the grid.
2. By setting the given values to the appropriate cells, these values should be removed from the domain of the peers (domain pruning).
3. The above step is repeated for all the cells in the input that have fixed values.
4. Once this has been done, there are three cases possible:
    a. The domain of some of the variables reduce to a single value.
    b. All the variables have at least 2 domains, hence use value based look up to assign new values and further prune the domain.
    c. Use the Naked Twins rule to eliminate the domain of cells that form a Naked Twin from its peers.
5. Step 4 is repeated until no further steps can be made. If the constraints enforced in this step is enough to reach the goal state, the problem is solved, and solution is returned. If no solution is returned, move to Step 6.

6. Backtracking is introduced with heuristics like MRV for variable ordering and LCV for value ordering.
7. The variable with the least domain size is chosen for assigning a value inside backtrack(MRV). The value for the chosen variable is taken to be the one that prunes the least values in its peer's domains.
8. Once the value is chosen, forward checking, arc consistency is performed.
9. To further reduce the domain size, value based lookup is also performed inside backtrack.
10. If forward checking fails (empty domain for some variables), the previous assignments are reverted and a different value for the variable is chosen.
11. Steps 7 to 10 is repeated until a solution is found.

Experimentation Results: -

The Algorithm was run on the benchmarks and the results are given below: -

Test	Time (On Average of 100 runs)	Best Time	Worst Time	No. of Backtracks
Hex 101	2.3809s	2.2725s	2.4124s	13
Hex 100	2.3564s	2.2542s	2.4059s	0
Hex 82	2.3821s	2.2816s	2.4286s	81
Hex 81	0.5473s	0.4821s	0.5631s	6
Hex 80	0.4905s	0.4642s	0.5321s	7
Hex 51	0.2539s	0.2158s	0.2890s	0
Hex 21	0.4690s	0.4377s	0.4822s	2
Hex 20	0.2045s	0.1845s	0.2137s	0

We initially had an algorithm that did the backtracking without Value based lookup inside it. Hence without this feature in the backtracking, we observed that the number of backtracks for the Hex 101 reached up to 650,000. The

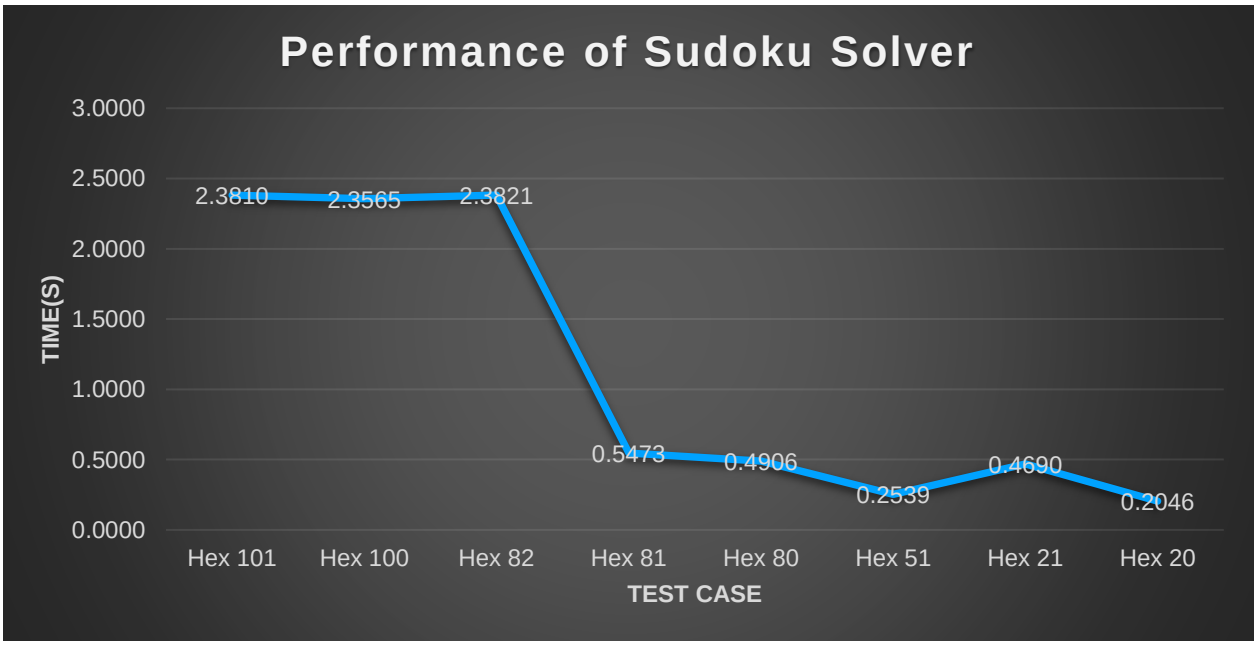
performance of the algorithm was also affected with the time taking more than 60s to find a solution for this test case.

Once we implemented the Value based lookup inside the Backtracking, we found that there was a complete drop in the number of backtracks that were being done. It reduced from over 600,000 to just 13 backtracks. This algorithm is the reason why we were able to get a better performance for our sudoku solver.

The reason that the Hex100 had 0 backtracks is that the initial assignment was a correct one and further selections were also correct with the help of LCV and MRV and thus we did not make one incorrect assignment and hence there was no need to backtrack.

The Hex51 and Hex20 on the other hand, did not enter backtracking as they were solved using the initial methods of Value based Lookup and Assigning single values.

Thus, with the help of LCV and MRV plus Backtracking using Value based Lookup, we were able to efficiently solve the Hexadecimal sudoku puzzles



Observations: -

- The code was written in Python 2.7 on a system with Intel Core i7 with 16GB RAM
- Our algorithm was able to solve all the benchmarks given above plus all the puzzles from sudoku-puzzles-online.com, of which their most difficult puzzle ran in 1.05s.
- For easy puzzles, we saw that there was no or less need of backtracking as the puzzle was solved by Value based lookup.
- In medium difficulty problems, the number of backtracks was under 10 and the number of state-space nodes generated was under 50.
- For Hard problems, the number of backtracks increased and was at a maximum of 100 with a state space of almost 200. Hence with this increase in the number of backtracks, the time also increased.

Further Improvements: -

- Further improvements could be made by porting the entire logic into C++ where we can expect even faster results than in Python.
- We can also implement parallelism in the pruning and un-pruning that we do when we backtrack, during initialization of various parameters for the grid and hence further reduce the time taken to compute the solution.
- Implementing a Minheap or a Counting sort to further reduce the sorting complexity that we use for MRV.

Appendix: -

Below is the output, time taken for the benchmarks, along with the search space nodes and number of backtracks.

HEX 20

```
Backtrack Count =  0
State Space Generated =  0
 -----------+-----------+-----------+-----------
 |  1 2 0 4  |  B F 6 9  |  C D E A  |  8 3 7 5  |
 |  3 E C B  |  8 D A 1  |  F 0 5 7  |  2 4 6 9  |
 |  A 5 9 7  |  2 3 E 4  |  6 1 8 B  |  D 0 F C  |
 |  8 F D 6  |  C 0 7 5  |  3 9 2 4  |  1 B A E  |
 -----------+-----------+-----------+-----------
 |  6 B 8 2  |  7 1 F E  |  9 3 A D  |  4 5 C 0  |
 |  4 A 7 1  |  0 9 5 8  |  2 E 6 C  |  3 D B F  |
 |  E 9 3 5  |  4 6 C D  |  B F 0 8  |  A 1 2 7  |
 |  0 D F C  |  3 B 2 A  |  1 4 7 5  |  6 9 E 8  |
 -----------+-----------+-----------+-----------
 |  B C 4 E  |  A 2 D F  |  8 5 1 3  |  9 7 0 6  |
 |  7 3 6 F  |  9 E 8 0  |  D A B 2  |  5 C 1 4  |
 |  5 8 A 9  |  6 4 1 3  |  0 7 C F  |  B E D 2  |
 |  2 0 1 D  |  5 C B 7  |  E 6 4 9  |  F A 8 3  |
 -----------+-----------+-----------+-----------
 |  C 6 2 3  |  E A 0 B  |  4 8 9 1  |  7 F 5 D  |
 |  D 7 B 8  |  F 5 9 6  |  A C 3 E  |  0 2 4 1  |
 |  9 1 E 0  |  D 7 4 2  |  5 B F 6  |  C 8 3 A  |
 |  F 4 5 A  |  1 8 3 C  |  7 2 D 0  |  E 6 9 B  |
 -----------+-----------+-----------+-----------
```

Time taken = 0.2045s

['1', '2', '0', '4', 'B', 'F', '6', '9', 'C', 'D', 'E', 'A', '8', '3', '7', '5', '3', 'E', 'C', 'B', '8', 'D', 'A', '1', 'F', '0', '5', '7', '2', '4', '6', '9', 'A', '5', '9', '7', '2', '3', 'E', '4', '6', '1', '8', 'B', 'D', '0', 'F', 'C', '8', 'F', 'D', '6', 'C', '0', '7', '5', '3', '9', '2', '4', '1', 'B', 'A', 'E', '6', 'B', '8', '2', '7', '1', 'F', 'E', '9', '3', 'A', 'D', '4', '5', 'C', '0', '4', 'A', '7', '1', '0', '9', '5', '8', '2', 'E', '6', 'C', '3', 'D', 'B', 'F', 'E', '9', '3', '5', '4', '6', 'C', 'D', 'B', 'F', '0', '8', 'A', '1', '2', '7', '0', 'D', 'F', 'C', '3', 'B', '2', 'A', '1', '4', '7', '5', '6', '9', 'E', '8', 'B', 'C', '4', 'E', 'A', '2', 'D', 'F', '8', '5', '1', '3', '9', '7', '0', '6', '7', '3', '6', 'F', '9', 'E', '8', '0', 'D', 'A', 'B', '2', '5', 'C', '1', '4', '5', '8', 'A', '9', '6', '4', '1', '3', '0', '7', 'C', 'F', 'B', 'E', 'D', '2', '2', '0', '1', 'D', '5', 'C', 'B', '7', 'E', '6', '4', '9', 'F', 'A', '8', '3', 'C', '6', '2', '3', 'E', 'A', '0', 'B', '4', '8', '9', '1', '7', 'F', '5', 'D', 'D', '7', 'B', '8', 'F', '5', '9', '6', 'A', 'C', '3', 'E', '0', '2', '4', '1', '9', '1', 'E', '0', 'D', '7', '4', '2', '5', 'B', 'F', '6', 'C', '8', '3', 'A', 'F', '4', '5', 'A', '1', '8', '3', 'C', '7', '2', 'D', '0', 'E', '6', '9', 'B']

HEX 21

```
Backtrack Count =  2
State Space Generated =  8
 -----------+-----------+-----------+-----------
 |  A 7 6 3  |  2 5 9 F  |  B C 1 0  |  8 4 D E  |
 |  C E 9 F  |  B 7 0 4  |  6 D 3 8  |  A 2 1 5  |
 |  2 0 D B  |  3 8 1 6  |  A 4 5 E  |  F 7 9 C  |
 |  8 1 4 5  |  D E A C  |  7 F 9 2  |  6 0 3 B  |
 -----------+-----------+-----------+-----------
 |  D F C 9  |  A 4 E 1  |  5 2 0 3  |  B 6 7 8  |
 |  4 A 5 6  |  F 3 8 9  |  C 7 B D  |  0 E 2 1  |
 |  1 3 0 2  |  7 6 B 5  |  F 8 E A  |  9 D C 4  |
 |  7 B 8 E  |  0 2 C D  |  9 1 4 6  |  5 A F 3  |
 -----------+-----------+-----------+-----------
 |  3 2 A 1  |  8 9 4 7  |  E B D F  |  C 5 0 6  |
 |  0 9 B 4  |  E C 6 3  |  1 5 8 7  |  D F A 2  |
 |  F 5 E C  |  1 0 D B  |  2 A 6 9  |  4 3 8 7  |
 |  6 8 7 D  |  5 F 2 A  |  3 0 C 4  |  1 B E 9  |
 -----------+-----------+-----------+-----------
 |  B 6 1 0  |  9 D F E  |  8 3 2 5  |  7 C 4 A  |
 |  E C 2 A  |  6 1 7 0  |  4 9 F B  |  3 8 5 D  |
 |  9 D 3 7  |  4 B 5 8  |  0 E A C  |  2 1 6 F  |
 |  5 4 F 8  |  C A 3 2  |  D 6 7 1  |  E 9 B 0  |
 -----------+-----------+-----------+-----------
```

Time taken = 0.4690s

['A', '7', '6', '3', '2', '5', '9', 'F', 'B', 'C', '1', '0', '8', '4', 'D', 'E', 'C', 'E', '9', 'F', 'B', '7', '0', '4', '6', 'D', '3', '8', 'A', '2', '1', '5', '2', '0', 'D', 'B', '3', '8', '1', '6', 'A', '4', '5', 'E', 'F', '7', '9', 'C', '8', '1', '4', '5', 'D', 'E', 'A', 'C', '7', 'F', '9', '2', '6', '0', '3', 'B', 'D', 'F', 'C', '9', 'A', '4', 'E', '1', '5', '2', '0', '3', 'B', '6', '7', '8', '4', 'A', '5', '6', 'F', '3', '8', '9', 'C', '7', 'B', 'D', '0', 'E', '2', '1', '1', '3', '0', '2', '7', '6', 'B', '5', 'F', '8', 'E', 'A', '9', 'D', 'C', '4', '7', 'B', '8', 'E', '0', '2', 'C', 'D', '9', '1', '4', '6', '5', 'A', 'F', '3', '3', '2', 'A', '1', '8', '9', '4', '7', 'E', 'B', 'D', 'F', 'C', '5', '0', '6', '0', '9', 'B', '4', 'E', 'C', '6', '3', '1', '5', '8', '7', 'D', 'F', 'A', '2', 'F', '5', 'E', 'C', '1', '0', 'D', 'B', '2', 'A', '6', '9', '4', '3', '8', '7', '6', '8', '7', 'D', '5', 'F', '2', 'A', '3', '0', 'C', '4', '1', 'B', 'E', '9', 'B', '6', '1', '0', '9', 'D', 'F', 'E', '8', '3', '2', '5', '7', 'C', '4', 'A', 'E', 'C', '2', 'A', '6', '1', '7', '0', '4', '9', 'F', 'B', '3', '8', '5', 'D', '9', 'D', '3', '7', '4', 'B', '5', '8', '0', 'E', 'A', 'C', '2', '1', '6', 'F', '5', '4', 'F', '8', 'C', 'A', '3', '2', 'D', '6', '7', '1', 'E', '9', 'B', '0']

HEX 51

```
Backtrack Count =  0
State Space Generated =  0
 -----------+-----------+-----------+-----------
 |  4 1 2 A  |  7 6 D 3  |  F 0 B 5  |  E 8 9 C |
 |  B 8 3 D  |  5 F C E  |  7 1 6 9  |  A 0 2 4 |
 |  7 F 5 E  |  0 4 9 A  |  C 3 2 8  |  D 6 1 B |
 |  0 9 C 6  |  B 2 8 1  |  D 4 E A  |  5 F 3 7 |
 -----------+-----------+-----------+-----------
 |  A 6 1 F  |  C E 3 5  |  4 9 D 7  |  0 B 8 2 |
 |  3 7 8 C  |  2 D 6 9  |  0 A 5 B  |  4 1 E F |
 |  D 4 0 5  |  8 1 F B  |  2 6 C E  |  9 7 A 3 |
 |  E B 9 2  |  4 A 0 7  |  3 8 1 F  |  6 C D 5 |
 -----------+-----------+-----------+-----------
 |  1 3 F 7  |  E 9 5 8  |  B D 0 4  |  2 A C 6 |
 |  2 C B 9  |  A 7 4 0  |  5 E 8 6  |  3 D F 1 |
 |  6 D E 4  |  1 3 B F  |  A 2 7 C  |  8 5 0 9 |
 |  8 5 A 0  |  D C 2 6  |  9 F 3 1  |  B 4 7 E |
 -----------+-----------+-----------+-----------
 |  C E D B  |  9 0 A 4  |  6 7 F 2  |  1 3 5 8 |
 |  5 A 4 1  |  3 B 7 2  |  8 C 9 0  |  F E 6 D |
 |  9 0 6 3  |  F 8 E C  |  1 5 4 D  |  7 2 B A |
 |  F 2 7 8  |  6 5 1 D  |  E B A 3  |  C 9 4 0 |
 -----------+-----------+-----------+-----------
```

Time taken = 0.2539s

['4', '1', '2', 'A', '7', '6', 'D', '3', 'F', '0', 'B', '5', 'E', '8', '9', 'C', 'B', '8', '3', 'D', '5', 'F', 'C', 'E', '7', '1', '6', '9', 'A', '0', '2', '4', '7', 'F', '5', 'E', '0', '4', '9', 'A', 'C', '3', '2', '8', 'D', '6', '1', 'B', '0', '9', 'C', '6', 'B', '2', '8', '1', 'D', '4', 'E', 'A', '5', 'F', '3', '7', 'A', '6', '1', 'F', 'C', 'E', '3', '5', '4', '9', 'D', '7', '0', 'B', '8', '2', '3', '7', '8', 'C', '2', 'D', '6', '9', '0', 'A', '5', 'B', '4', '1', 'E', 'F', 'D', '4', '0', '5', '8', '1', 'F', 'B', '2', '6', 'C', 'E', '9', '7', 'A', '3', 'E', 'B', '9', '2', '4', 'A', '0', '7', '3', '8', '1', 'F', '6', 'C', 'D', '5', '1', '3', 'F', '7', 'E', '9', '5', '8', 'B', 'D', '0', '4', '2', 'A', 'C', '6', '2', 'C', 'B', '9', 'A', '7', '4', '0', '5', 'E', '8', '6', '3', 'D', 'F', '1', '6', 'D', 'E', '4', '1', '3', 'B', 'F', 'A', '2', '7', 'C', '8', '5', '0', '9', '8', '5', 'A', '0', 'D', 'C', '2', '6', '9', 'F', '3', '1', 'B', '4', '7', 'E', 'C', 'E', 'D', 'B', '9', '0', 'A', '4', '6', '7', 'F', '2', '1', '3', '5', '8', '5', 'A', '4', '1', '3', 'B', '7', '2', '8', 'C', '9', '0', 'F', 'E', '6', 'D', '9', '0', '6', '3', 'F', '8', 'E', 'C', '1', '5', '4', 'D', '7', '2', 'B', 'A', 'F', '2', '7', '8', '6', '5', '1', 'D', 'E', 'B', 'A', '3', 'C', '9', '4', '0']

HEX 80

```
Backtrack Count =  7
State Space Generated =  17
 -----------+-----------+-----------+-----------
 |  A E F 0  |  5 3 9 2  |  1 8 C B  |  D 6 7 4 |
 |  3 9 D 4  |  C B E 6  |  F 2 5 7  |  A 1 0 8 |
 |  6 C 5 8  |  1 4 0 7  |  A D 9 E  |  2 B 3 F |
 |  1 7 2 B  |  A D F 8  |  6 3 4 0  |  9 E 5 C |
 -----------+-----------+-----------+-----------
 |  9 5 8 6  |  E 2 7 F  |  C B 1 A  |  3 D 4 0 |
 |  B F C E  |  9 6 5 A  |  D 0 3 4  |  8 7 2 1 |
 |  4 2 1 A  |  3 0 C D  |  5 7 6 8  |  F 9 E B |
 |  7 0 3 D  |  8 1 B 4  |  2 F E 9  |  C 5 A 6 |
 -----------+-----------+-----------+-----------
 |  2 B 4 C  |  6 E 3 5  |  9 1 7 D  |  0 F 8 A |
 |  E A 9 3  |  2 8 D C  |  B 6 0 F  |  5 4 1 7 |
 |  F 1 0 5  |  B 7 A 9  |  4 E 8 2  |  6 C D 3 |
 |  8 D 6 7  |  4 F 1 0  |  3 5 A C  |  B 2 9 E |
 -----------+-----------+-----------+-----------
 |  C 3 E 1  |  D A 6 B  |  8 4 2 5  |  7 0 F 9 |
 |  5 6 7 9  |  F C 4 E  |  0 A D 3  |  1 8 B 2 |
 |  0 8 A F  |  7 5 2 1  |  E 9 B 6  |  4 3 C D |
 |  D 4 B 2  |  0 9 8 3  |  7 C F 1  |  E A 6 5 |
 -----------+-----------+-----------+-----------
```

Time taken = 0.4905s

['A', 'E', 'F', '0', '5', '3', '9', '2', '1', '8', 'C', 'B', 'D', '6', '7', '4', '3', '9', 'D', '4', 'C', 'B', 'E', '6', 'F', '2', '5', '7', 'A', '1', '0', '8', '6', 'C', '5', '8', '1', '4', '0', '7', 'A', 'D', '9', 'E', '2', 'B', '3', 'F', '1', '7', '2', 'B', 'A', 'D', 'F', '8', '6', '3', '4', '0', '9', 'E', '5', 'C', '9', '5', '8', '6', 'E', '2', '7', 'F', 'C', 'B', '1', 'A', '3', 'D', '4', '0', 'B', 'F', 'C', 'E', '9', '6', '5', 'A', 'D', '0', '3', '4', '8', '7', '2', '1', '4', '2', '1', 'A', '3', '0', 'C', 'D', '5', '7', '6', '8', 'F', '9', 'E', 'B', '7', '0', '3', 'D', '8', '1', 'B', '4', '2', 'F', 'E', '9', 'C', '5', 'A', '6', '2', 'B', '4', 'C', '6', 'E', '3', '5', '9', '1', '7', 'D', '0', 'F', '8', 'A', 'E', 'A', '9', '3', '2', '8', 'D', 'C', 'B', '6', '0', 'F', '5', '4', '1', '7', 'F', '1', '0', '5', 'B', '7', 'A', '9', '4', 'E', '8', '2', '6', 'C', 'D', '3', '8', 'D', '6', '7', '4', 'F', '1', '0', '3', '5', 'A', 'C', 'B', '2', '9', 'E', 'C', '3', 'E', '1', 'D', 'A', '6', 'B', '8', '4', '2', '5', '7', '0', 'F', '9', '5', '6', '7', '9', 'F', 'C', '4', 'E', '0', 'A', 'D', '3', '1', '8', 'B', '2', '0', '8', 'A', 'F', '7', '5', '2', '1', 'E', '9', 'B', '6', '4', '3', 'C', 'D', 'D', '4', 'B', '2', '0', '9', '8', '3', '7', 'C', 'F', '1', 'E', 'A', '6', '5']

HEX 81

```
Backtrack Count =  6
State Space Generated =  18
 -----------+-----------+-----------+-----------
 |  A 9 3 7  |  4 D 1 2  |  8 F 6 B  |  E 5 0 C  |
 |  6 0 E F  |  8 9 B A  |  5 2 7 C  |  D 1 4 3  |
 |  D 2 1 B  |  3 6 C 5  |  E 0 4 A  |  7 F 8 9  |
 |  C 4 8 5  |  E 0 F 7  |  1 3 9 D  |  6 B A 2  |
 -----------+-----------+-----------+-----------
 |  8 B 9 E  |  0 3 7 4  |  C 5 D F  |  2 A 6 1  |
 |  F 1 D C  |  5 A 2 6  |  0 7 B 9  |  3 8 E 4  |
 |  2 5 4 0  |  F 1 8 D  |  3 6 A E  |  9 7 C B  |
 |  7 3 A 6  |  C E 9 B  |  2 4 1 8  |  5 D F 0  |
 -----------+-----------+-----------+-----------
 |  4 7 5 2  |  A F 0 E  |  D 8 3 1  |  B C 9 6  |
 |  0 E B A  |  6 8 D C  |  7 9 F 4  |  1 2 3 5  |
 |  3 D F 1  |  2 B 5 9  |  6 A C 0  |  4 E 7 8  |
 |  9 6 C 8  |  7 4 3 1  |  B E 5 2  |  A 0 D F  |
 -----------+-----------+-----------+-----------
 |  E 8 7 3  |  D C 6 F  |  4 B 2 5  |  0 9 1 A  |
 |  1 A 6 4  |  B 5 E 0  |  9 C 8 7  |  F 3 2 D  |
 |  B C 2 9  |  1 7 A 3  |  F D 0 6  |  8 4 5 E  |
 |  5 F 0 D  |  9 2 4 8  |  A 1 E 3  |  C 6 B 7  |
 -----------+-----------+-----------+-----------
```

Time taken = 0.5473s

['A', '9', '3', '7', '4', 'D', '1', '2', '8', 'F', '6', 'B', 'E', '5', '0', 'C', '6', '0', 'E', 'F', '8', '9', 'B', 'A', '5', '2', '7', 'C', 'D', '1', '4', '3', 'D', '2', '1', 'B', '3', '6', 'C', '5', 'E', '0', '4', 'A', '7', 'F', '8', '9', 'C', '4', '8', '5', 'E', '0', 'F', '7', '1', '3', '9', 'D', '6', 'B', 'A', '2', '8', 'B', '9', 'E', '0', '3', '7', '4', 'C', '5', 'D', 'F', '2', 'A', '6', '1', 'F', '1', 'D', 'C', '5', 'A', '2', '6', '0', '7', 'B', '9', '3', '8', 'E', '4', '2', '5', '4', '0', 'F', '1', '8', 'D', '3', '6', 'A', 'E', '9', '7', 'C', 'B', '7', '3', 'A', '6', 'C', 'E', '9', 'B', '2', '4', '1', '8', '5', 'D', 'F', '0', '4', '7', '5', '2', 'A', 'F', '0', 'E', 'D', '8', '3', '1', 'B', 'C', '9', '6', '0', 'E', 'B', 'A', '6', '8', 'D', 'C', '7', '9', 'F', '4', '1', '2', '3', '5', '3', 'D', 'F', '1', '2', 'B', '5', '9', '6', 'A', 'C', '0', '4', 'E', '7', '8', '9', '6', 'C', '8', '7', '4', '3', '1', 'B', 'E', '5', '2', 'A', '0', 'D', 'F', 'E', '8', '7', '3', 'D', 'C', '6', 'F', '4', 'B', '2', '5', '0', '9', '1', 'A', '1', 'A', '6', '4', 'B', '5', 'E', '0', '9', 'C', '8', '7', 'F', '3', '2', 'D', 'B', 'C', '2', '9', '1', '7', 'A', '3', 'F', 'D', '0', '6', '8', '4', '5', 'E', '5', 'F', '0', 'D', '9', '2', '4', '8', 'A', '1', 'E', '3', 'C', '6', 'B', '7']

HEX 82

```
Backtrack Count =  81
State Space Generated =  103
 -----------+-----------+-----------+-----------
 |  8 D 6 2  |  A 5 C 7  |  E 4 0 3  |  F 1 B 9  |
 |  1 7 4 3  |  8 2 D 9  |  B 6 A F  |  0 5 E C  |
 |  0 E 5 F  |  1 B 4 3  |  9 D 8 C  |  6 A 7 2  |
 |  B C A 9  |  6 E 0 F  |  2 1 7 5  |  8 D 4 3  |
 -----------+-----------+-----------+-----------
 |  5 F 9 B  |  7 6 A 2  |  3 C D 8  |  E 4 0 1  |
 |  D 6 E 4  |  C F 3 5  |  1 0 B 2  |  9 8 A 7  |
 |  7 2 0 A  |  D 1 8 E  |  F 9 4 6  |  B C 3 5  |
 |  C 8 3 1  |  B 4 9 0  |  7 A 5 E  |  2 F 6 D  |
 -----------+-----------+-----------+-----------
 |  E A 8 6  |  9 3 B D  |  C 2 F 0  |  5 7 1 4  |
 |  F 0 2 C  |  E 7 5 4  |  8 3 6 1  |  A 9 D B  |
 |  4 3 7 5  |  0 8 1 6  |  A B 9 D  |  C E 2 F  |
 |  9 1 B D  |  F C 2 A  |  4 5 E 7  |  3 6 8 0  |
 -----------+-----------+-----------+-----------
 |  2 9 C 8  |  5 D 7 B  |  0 E 1 A  |  4 3 F 6  |
 |  3 5 F E  |  4 0 6 8  |  D 7 2 9  |  1 B C A  |
 |  A 4 D 0  |  3 9 E 1  |  6 F C B  |  7 2 5 8  |
 |  6 B 1 7  |  2 A F C  |  5 8 3 4  |  D 0 9 E  |
 -----------+-----------+-----------+-----------
```

Time taken = 2.3821s

['8', 'D', '6', '2', 'A', '5', 'C', '7', 'E', '4', '0', '3', 'F', '1', 'B', '9', '1', '7', '4', '3', '8', '2', 'D', '9', 'B', '6', 'A', 'F', '0', '5', 'E', 'C', '0', 'E', '5', 'F', '1', 'B', '4', '3', '9', 'D', '8', 'C', '6', 'A', '7', '2', 'B', 'C', 'A', '9', '6', 'E', '0', 'F', '2', '1', '7', '5', '8', 'D', '4', '3', '5', 'F', '9', 'B', '7', '6', 'A', '2', '3', 'C', 'D', '8', 'E', '4', '0', '1', 'D', '6', 'E', '4', 'C', 'F', '3', '5', '1', '0', 'B', '2', '9', '8', 'A', '7', '7', '2', '0', 'A', 'D', '1', '8', 'E', 'F', '9', '4', '6', 'B', 'C', '3', '5', 'C', '8', '3', '1', 'B', '4', '9', '0', '7', 'A', '5', 'E', '2', 'F', '6', 'D', 'E', 'A', '8', '6', '9', '3', 'B', 'D', 'C', '2', 'F', '0', '5', '7', '1', '4', 'F', '0', '2', 'C', 'E', '7', '5', '4', '8', '3', '6', '1', 'A', '9', 'D', 'B', '4', '3', '7', '5', '0', '8', '1', '6', 'A', 'B', '9', 'D', 'C', 'E', '2', 'F', '9', '1', 'B', 'D', 'F', 'C', '2', 'A', '4', '5', 'E', '7', '3', '6', '8', '0', '2', '9', 'C', '8', '5', 'D', '7', 'B', '0', 'E', '1', 'A', '4', '3', 'F', '6', '3', '5', 'F', 'E', '4', '0', '6', '8', 'D', '7', '2', '9', '1', 'B', 'C', 'A', 'A', '4', 'D', '0', '3', '9', 'E', '1', '6', 'F', 'C', 'B', '7', '2', '5', '8', '6', 'B', '1', '7', '2', 'A', 'F', 'C', '5', '8', '3', '4', 'D', '0', '9', 'E']

HEX 100

```
Backtrack Count =  0
State Space Generated =  133
+-----------+-----------+-----------+-----------+
|  6 D 4 5  |  E 8 2 1  |  7 9 F A  |  3 0 C B  |
|  E B 1 8  |  A C 5 7  |  3 0 D 2  |  F 6 4 9  |
|  A 0 9 C  |  D 3 F B  |  6 E 5 4  |  1 8 7 2  |
|  2 3 F 7  |  0 6 4 9  |  1 B C 8  |  E A 5 D  |
+-----------+-----------+-----------+-----------+
|  D F A 4  |  5 0 6 2  |  E 8 9 1  |  7 3 B C  |
|  0 C 6 B  |  8 E D 3  |  5 A 4 7  |  2 F 9 1  |
|  7 2 5 9  |  1 F B C  |  D 3 6 0  |  A 4 8 E  |
|  3 8 E 1  |  7 9 A 4  |  C 2 B F  |  D 5 6 0  |
+-----------+-----------+-----------+-----------+
|  8 4 3 6  |  F 2 E D  |  A 5 1 9  |  C B 0 7  |
|  5 1 0 A  |  9 B 3 6  |  2 C 7 E  |  4 D F 8  |
|  C E B F  |  4 A 7 5  |  0 D 8 6  |  9 2 1 3  |
|  9 7 2 D  |  C 1 0 8  |  4 F 3 B  |  5 E A 6  |
+-----------+-----------+-----------+-----------+
|  F 6 D E  |  B 7 1 A  |  9 4 0 3  |  8 C 2 5  |
|  4 A C 0  |  6 5 9 E  |  8 7 2 D  |  B 1 3 F  |
|  1 9 8 3  |  2 4 C F  |  B 6 E 5  |  0 7 D A  |
|  B 5 7 2  |  3 D 8 0  |  F 1 A C  |  6 9 E 4  |
+-----------+-----------+-----------+-----------+
```

Time taken = 2.3565s

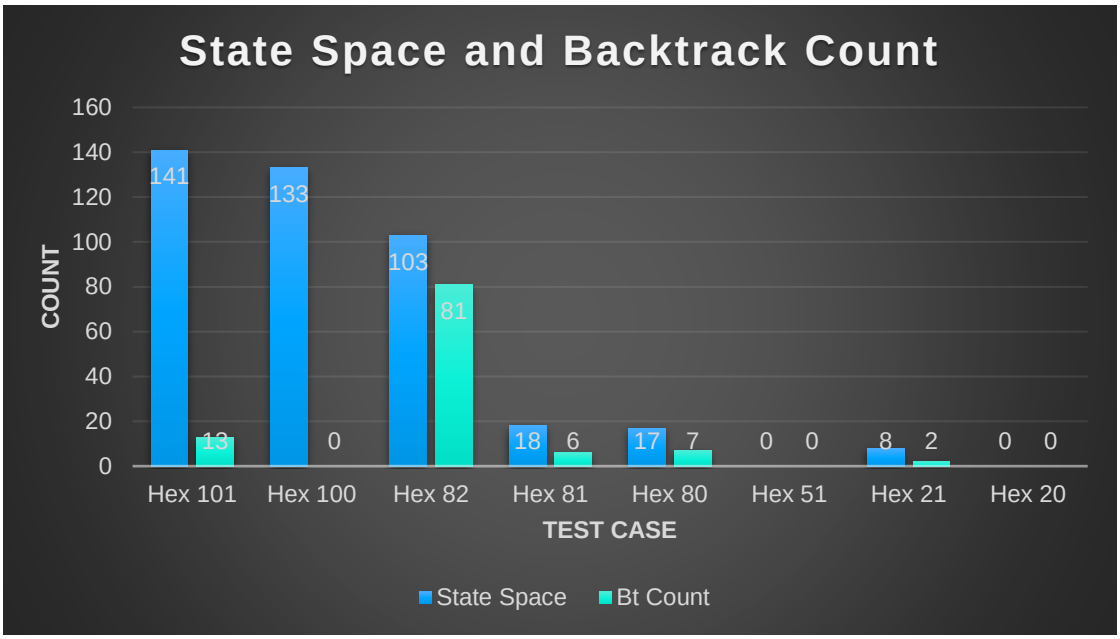
['6', 'D', '4', '5', 'E', '8', '2', '1', '7', '9', 'F', 'A', '3', '0', 'C', 'B', 'E', 'B', '1', '8', 'A', 'C', '5', '7', '3', '0', 'D', '2', 'F', '6', '4', '9', 'A', '0', '9', 'C', 'D', '3', 'F', 'B', '6', 'E', '5', '4', '1', '8', '7', '2', '2', '3', 'F', '7', '0', '6', '4', '9', '1', 'B', 'C', '8', 'E', 'A', '5', 'D', 'D', 'F', 'A', '4', '5', '0', '6', '2', 'E', '8', '9', '1', '7', '3', 'B', 'C', '0', 'C', '6', 'B', '8', 'E', 'D', '3', '5', 'A', '4', '7', '2', 'F', '9', '1', '7', '2', '5', '9', '1', 'F', 'B', 'C', 'D', '3', '6', '0', 'A', '4', '8', 'E', '3', '8', 'E', '1', '7', '9', 'A', '4', 'C', '2', 'B', 'F', 'D', '5', '6', '0', '8', '4', '3', '6', 'F', '2', 'E', 'D', 'A', '5', '1', '9', 'C', 'B', '0', '7', '5', '1', '0', 'A', '9', 'B', '3', '6', '2', 'C', '7', 'E', '4', 'D', 'F', '8', 'C', 'E', 'B', 'F', '4', 'A', '7', '5', '0', 'D', '8', '6', '9', '2', '1', '3', '9', '7', '2', 'D', 'C', '1', '0', '8', '4', 'F', '3', 'B', '5', 'E', 'A', '6', 'F', '6', 'D', 'E', 'B', '7', '1', 'A', '9', '4', '0', '3', '8', 'C', '2', '5', '4', 'A', 'C', '0', '6', '5', '9', 'E', '8', '7', '2', 'D', 'B', '1', '3', 'F', '1', '9', '8', '3', '2', '4', 'C', 'F', 'B', '6', 'E', '5', '0', '7', 'D', 'A', 'B', '5', '7', '2', '3', 'D', '8', '0', 'F', '1', 'A', 'C', '6', '9', 'E', '4']

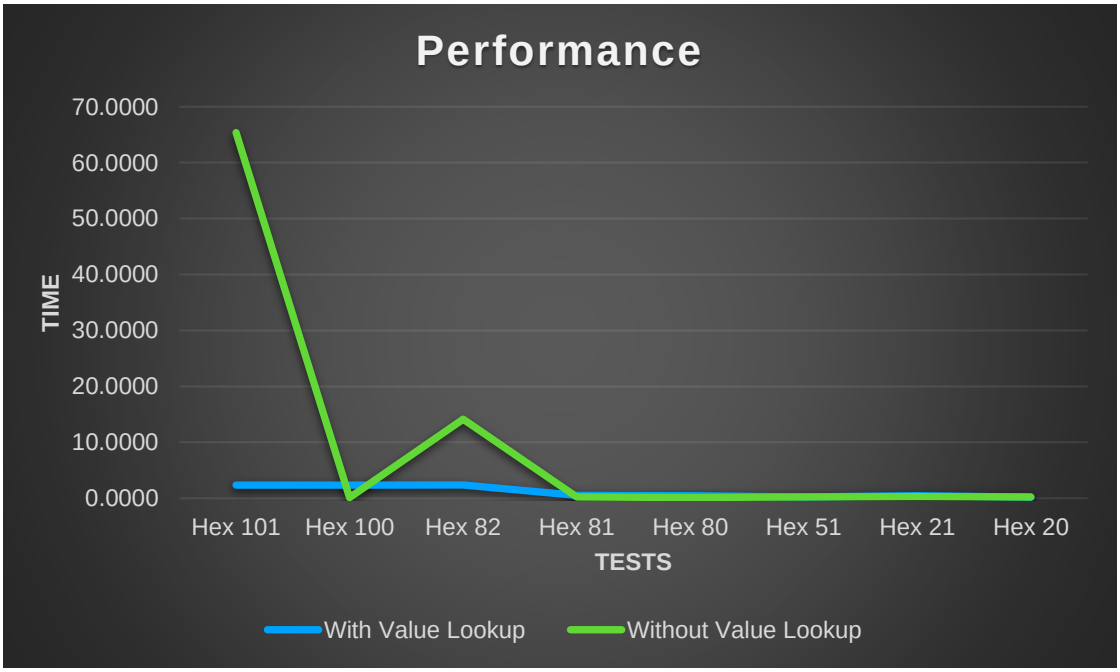
HEX 101

```
Backtrack Count =  13
State Space Generated =  141
+-----------+-----------+-----------+-----------+
|  4 7 A E  |  1 0 5 8  |  F 3 9 B  |  C 2 D 6  |
|  D F B 8  |  2 A 4 7  |  C 6 5 0  |  E 1 3 9  |
|  0 5 6 9  |  D C 3 E  |  4 1 2 8  |  F B A 7  |
|  C 3 1 2  |  F B 6 9  |  E D A 7  |  8 4 5 0  |
+-----------+-----------+-----------+-----------+
|  2 8 D 7  |  C 6 0 3  |  A B 1 9  |  5 F 4 E  |
|  B C 3 4  |  E D 9 5  |  7 F 6 2  |  A 0 1 8  |
|  9 A E 1  |  7 F 2 4  |  D 0 8 5  |  B 3 6 C  |
|  6 0 F 5  |  A 8 B 1  |  3 E C 4  |  9 D 7 2  |
+-----------+-----------+-----------+-----------+
|  5 2 9 D  |  8 7 F 6  |  B A 0 1  |  3 E C 4  |
|  3 E C A  |  4 9 D 0  |  5 7 F 6  |  2 8 B 1  |
|  8 6 4 B  |  5 E 1 C  |  9 2 D 3  |  0 7 F A  |
|  7 1 0 F  |  3 2 A B  |  8 4 E C  |  D 6 9 5  |
+-----------+-----------+-----------+-----------+
|  1 4 5 C  |  9 3 E D  |  2 8 7 F  |  6 A 0 B  |
|  F B 7 3  |  6 5 8 2  |  0 9 4 A  |  1 C E D  |
|  A D 2 6  |  0 4 C F  |  1 5 B E  |  7 9 8 3  |
|  E 9 8 0  |  B 1 7 A  |  6 C 3 D  |  4 5 2 F  |
+-----------+-----------+-----------+-----------+
```

Time taken = 2.3810s

['4', '7', 'A', 'E', '1', '0', '5', '8', 'F', '3', '9', 'B', 'C', '2', 'D', '6', 'D', 'F', 'B', '8', '2', 'A', '4', '7', 'C', '6', '5', '0', 'E', '1', '3', '9', '0', '5', '6', '9', 'D', 'C', '3', 'E', '4', '1', '2', '8', 'F', 'B', 'A', '7', 'C', '3', '1', '2', 'F', 'B', '6', '9', 'E', 'D', 'A', '7', '8', '4', '5', '0', '2', '8', 'D', '7', 'C', '6', '0', '3', 'A', 'B', '1', '9', '5', 'F', '4', 'E', 'B', 'C', '3', '4', 'E', 'D', '9', '5', '7', 'F', '6', '2', 'A', '0', '1', '8', '9', 'A', 'E', '1', '7', 'F', '2', '4', 'D', '0', '8', '5', 'B', '3', '6', 'C', '6', '0', 'F', '5', 'A', '8', 'B', '1', '3', 'E', 'C', '4', '9', 'D', '7', '2', '5', '2', '9', 'D', '8', '7', 'F', '6', 'B', 'A', '0', '1', '3', 'E', 'C', '4', '3', 'E', 'C', 'A', '4', '9', 'D', '0', '5', '7', 'F', '6', '2', '8', 'B', '1', '8', '6', '4', 'B', '5', 'E', '1', 'C', '9', '2', 'D', '3', '0', '7', 'F', 'A', '7', '1', '0', 'F', '3', '2', 'A', 'B', '8', '4', 'E', 'C', 'D', '6', '9', '5', '1', '4', '5', 'C', '9', '3', 'E', 'D', '2', '8', '7', 'F', '6', 'A', '0', 'B', 'F', 'B', '7', '3', '6', '5', '8', '2', '0', '9', '4', 'A', '1', 'C', 'E', 'D', 'A', 'D', '2', '6', '0', '4', 'C', 'F', '1', '5', 'B', 'E', '7', '9', '8', '3', 'E', '9', '8', '0', 'B', '1', '7', 'A', '6', 'C', '3', 'D', '4', '5', '2', 'F']


State Space and Backtrack Count
160
140
120
100
80
60
40
20
0
COUNT
141
13
133
0
103
81
18
6
17
7
0
0
8
2
0
0
Hex 101
Hex 100
Hex 82
Hex 81
Hex 80
Hex 51
Hex 21
Hex 20
TEST CASE
State Space
Bt Count


Performance
70.0000
60.0000
50.0000
40.0000
30.0000
20.0000
10.0000
0.0000
TIME
Hex 101
Hex 100
Hex 82
Hex 81
Hex 80
Hex 51
Hex 21
Hex 20
TESTS
With Value Lookup
Without Value Lookup