Artificial Intelligence HW-3 Report

By: Ayush Agarwal - 111416587, Soumyakant Priyadarshan-111580387

We have implemented the following CSP techniques to find the golomb ruler for a given length L and order M. For all the techniques, we have first found whether there is a possible golomb ruler for given length and order. If there exists a possible golomb ruler, process used to find optimal length is:-

1. L = given length
2. M= given order
3. While golomb ruler possible decrease L by 1
4. Optimal Length = L+1

Simple Backtracking approach:

Test result:

1) L = 6, M = 4
Answer 6,[0,1,4,6]
Number of recursions to find golomb ruler for given length and order: 5
Number of recursions to find optimal golomb ruler for given order: 14

2) L = 23, M=7
Answer -1,[]
Number of recursions to find golomb ruler for given length and order: 4052

3) L = 30, M = 7
Answer 25, [0, 1, 4, 10, 18, 23, 25]
Number of recursions to find golomb ruler for given length and order: 7
Number of recursions to find optimal golomb ruler for given order: 6195

4) L = 45, M=8
Answer 34, [0, 1, 4, 9, 15, 22, 32, 34]
Number of recursions to find golomb ruler for given length and order: 8
Number of recursions to find optimal golomb ruler for given order: 58032

5) L = 44, M=9
Answer 34, [0, 1, 4, 9, 15, 22, 32, 34]
Number of recursions to find golomb ruler for given length and order: 16221
Number of recursions to find optimal golomb ruler for given order: 422512

Forward Checking approach:

Test result:

1) L = 6, M = 4
Answer 6,[0,1,4,6]
Number of recursions to find golomb ruler for given length and order: 4

Artificial Intelligence HW-3 Report

By: Ayush Agarwal - 111416587, Soumyakant Priyadarshan-111580387

Number of recursions to find optimal golomb ruler for given order: 9

2) L = 23, M=7
Answer -1,[]
Number of recursions to find golomb ruler for given length and order: 1320

3) L = 30, M = 7
Answer 25, [0, 1, 4, 10, 18, 23, 25]
Number of recursions to find golomb ruler for given length and order: 7
Number of recursions to find optimal golomb ruler for given order: 1762

4) L = 45, M=8
Answer 34, [0, 1, 4, 9, 15, 22, 32, 34]
Number of recursions to find golomb ruler for given length and order: 8
Number of recursions to find optimal golomb ruler for given order: 15109

5) L = 44, M=9
Answer 34, [0, 1, 4, 9, 15, 22, 32, 34]
Number of recursions to find golomb ruler for given length and order: 3203
Number of recursions to find optimal golomb ruler for given order: 97828

Constraint propagation approach:

<u>Test result</u>:

1) L = 6, M = 4
Answer 6,[0,1,4,6]
Number of recursions to find golomb ruler for given length and order: 4
Number of recursions to find optimal golomb ruler for given order: 5

2) L = 23, M=7
Answer -1,[]
Number of recursions to find golomb ruler for given length and order: 419

3) L = 30, M = 7
Answer 25, [0, 1, 4, 10, 18, 23, 25]
Number of recursions to find golomb ruler for given length and order: 7
Number of recursions to find optimal golomb ruler for given order: 560

4) L = 45, M=8
Answer 34, [0, 1, 4, 9, 15, 22, 32, 34]
Number of recursions to find golomb ruler for given length and order: 8
Number of recursions to find optimal golomb ruler for given order: 4209

By: Ayush Agarwal - 111416587, Soumyakant Priyadarshan-111580387

5) L = 44, M=9
Answer 34, [0, 1, 4, 9, 15, 22, 32, 34]
Number of recursions to find golomb ruler for given length and order: 742
Number of recursions to find optimal golomb ruler for given order: 18488

Observation:

In general simple backtracking approach expands more nodes than forward checking and constraint propagation approach. The constraint propagation approach expands the least number of nodes. We also observed that each of the above algorithm undergo more number of recursions for the cases where no golomb ruler exists for given order and length. All the algorithms are taking considerable amount of time to find optimal length for order above 8 for the reason stated above(nearly 20 mins for order=9).
Where as it takes much lesser number of recursions and hence less time to find a valid golomb ruler(may not be optimal), if it exists.