CSP Review

CSP EXAMPLE: N-QUEEN

Outline

- Problem description for the N-Queen problem
- Backtracking Search (BTS) for the N-Queen problem
- Improving BTS for the N-Queen problem
- Summary

Outline

- **Problem description for the N-Queen problem**
- Backtracking Search (BTS) for the N-Queen problem
- Improving BTS for the N-Queen problem
- Summary

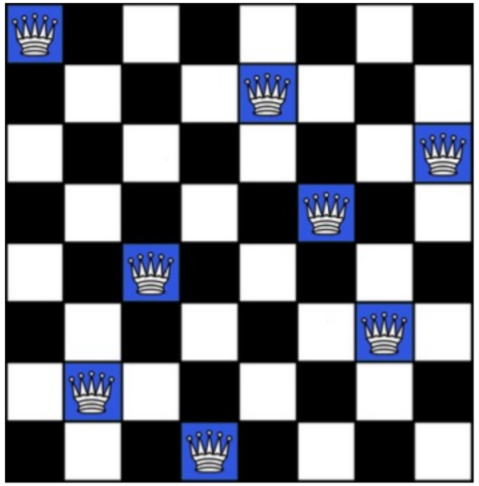
A brief Review of the CSP

- A Constraint Satisfaction Problem (CSP) consists of three elements:
  - A set of variables: X
  - A set of domains for each variable: D
  - A set of constraints C that specify allowable combinations of values

- Solving the CSP: finding the assignment(s) that satisfy all constraints.

The N-Queen Problem

- The N-Queen is the problem of placing N chess queens on an $N \times N$ chessboard so that no two queens attack each other.



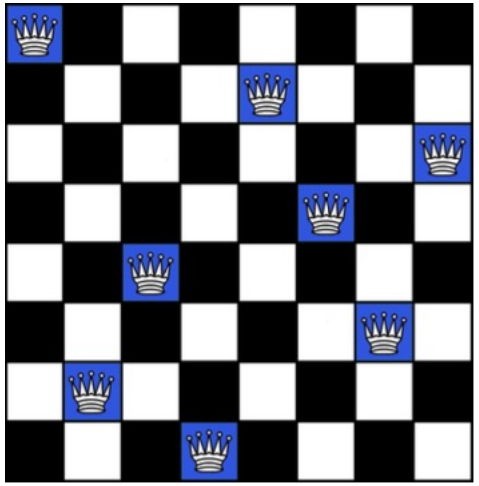
Variables: $X = \{(x_1, y_1), (x_2, y_2), \ldots (x_N, y_N)\}$

Domain: $D = \{1,2, \ldots N\}$

Constraints: $x_i \neq x_j, y_i \neq y_j, |x_i - x_j| \neq |y_i - y_j|, \forall i \neq j$

Solution Representation (1)

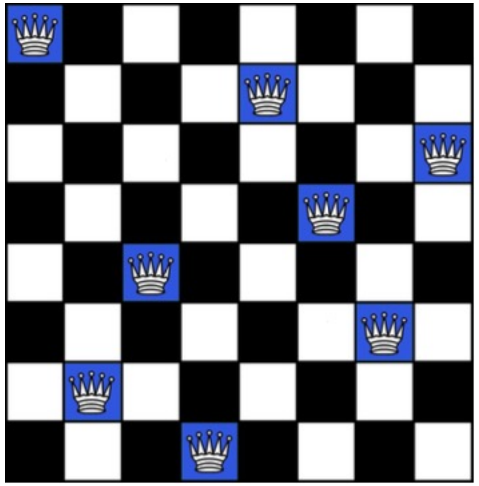
- One variable per queen, $Q_1, Q_2, \dots, Q_n$.
- Each variable could be a tuple (x, y), $\quad x, y \in [1, n]$.



$$\begin{aligned}
Q_1 &= (1,1)\\
Q_2 &= (2,5)\\
Q_3 &= (3,8)\\
Q_4 &= (4,6)\\
Q_5 &= (5,3)\\
Q_6 &= (6,7)\\
Q_7 &= (7,2)\\
Q_8 &= (8,4)
\end{aligned}$$

Solution Representation (2)

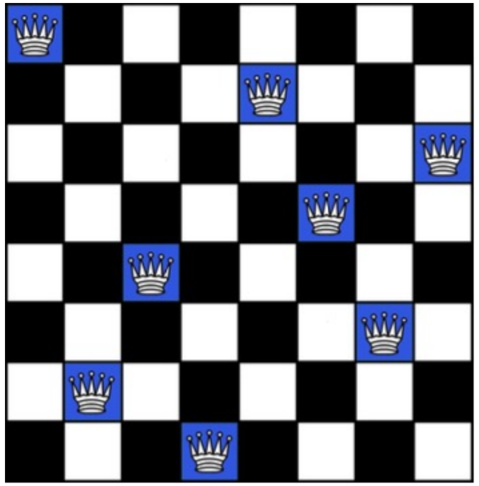
- One variable per queen, $Q_1, Q_2, \dots, Q_n$.
- Each variable could have a value ∈ $[1, n^2]$.



$$\begin{aligned} Q_1 &= 1 \\ Q_2 &= 13 \\ Q_3 &= 24 \\ Q_4 &= 30 \\ Q_5 &= 35 \\ Q_6 &= 47 \\ Q_7 &= 50 \\ Q_8 &= 60 \end{aligned}$$

Solution Representation (3)

- One variable per queen, $Q_1, Q_2, \dots, Q_n$.
- Each variable could have a value ∈ $[1, n]$.

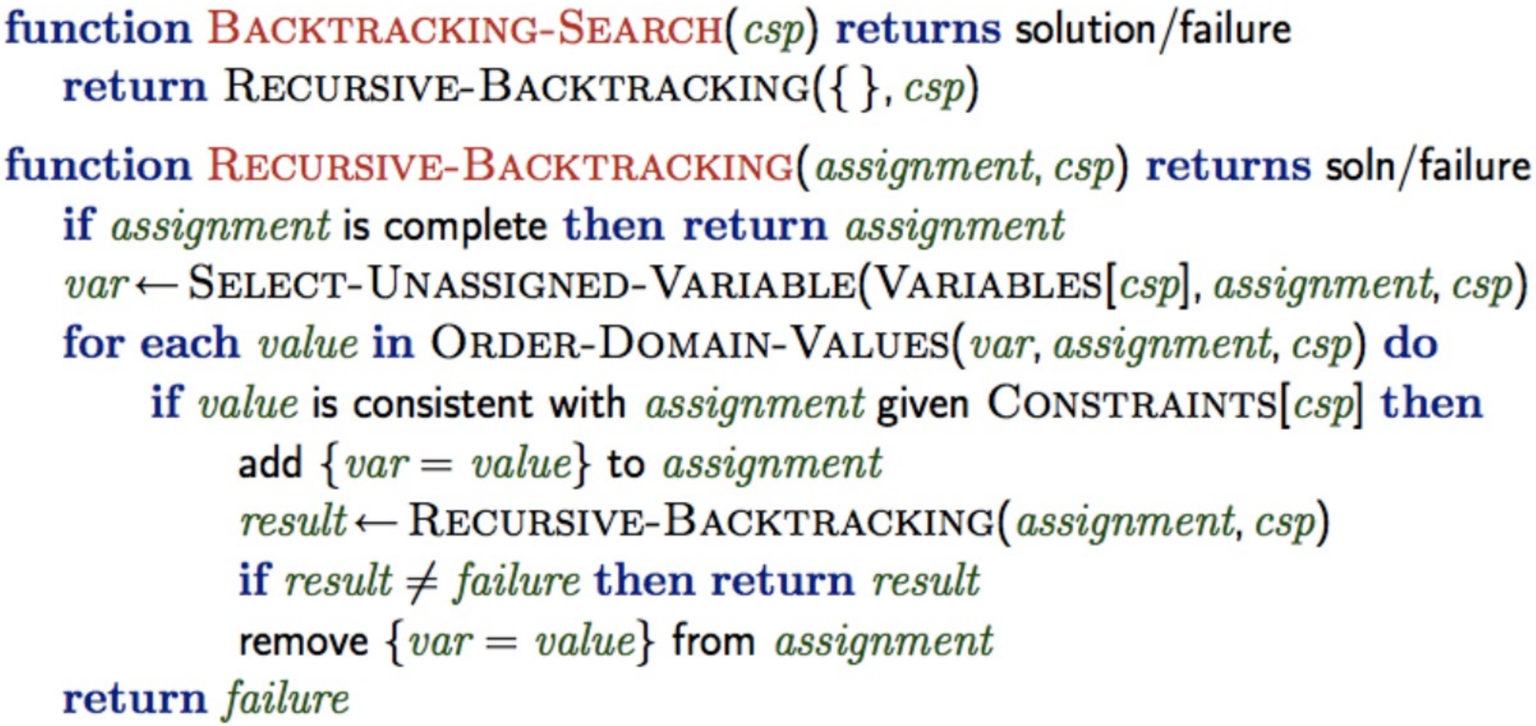


$$
\begin{aligned}
Q_1 &= 1\\
Q_2 &= 5\\
Q_3 &= 8\\
Q_4 &= 6\\
Q_5 &= 3\\
Q_6 &= 7\\
Q_7 &= 2\\
Q_8 &= 4
\end{aligned}
$$

Outline

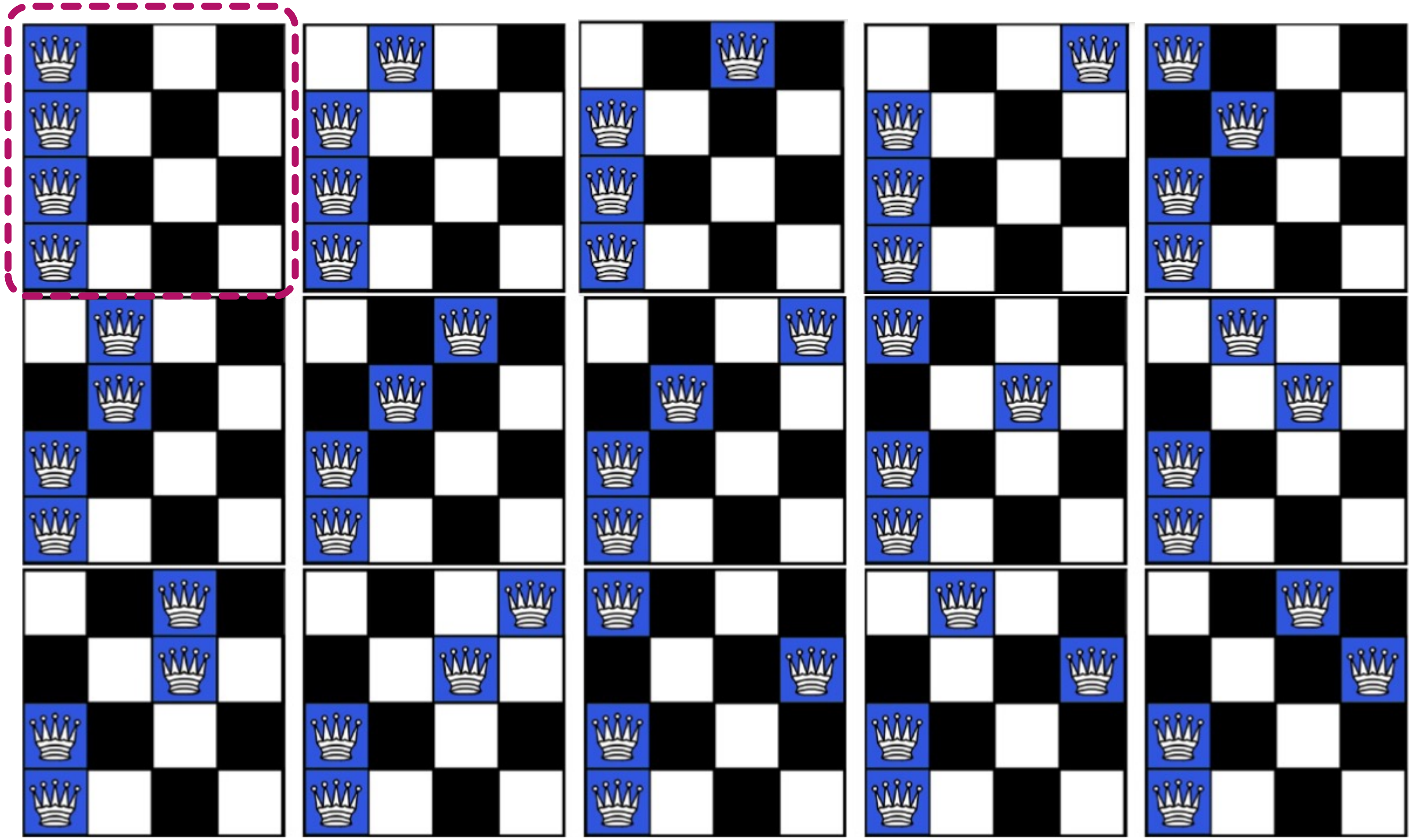
- Problem description for the N-Queen problem
- **Backtracking Search (BTS) for the N-Queen problem**
- Improving BTS for the N-Queen problem
- Summary

Backtracking Search(BTS) for CSP



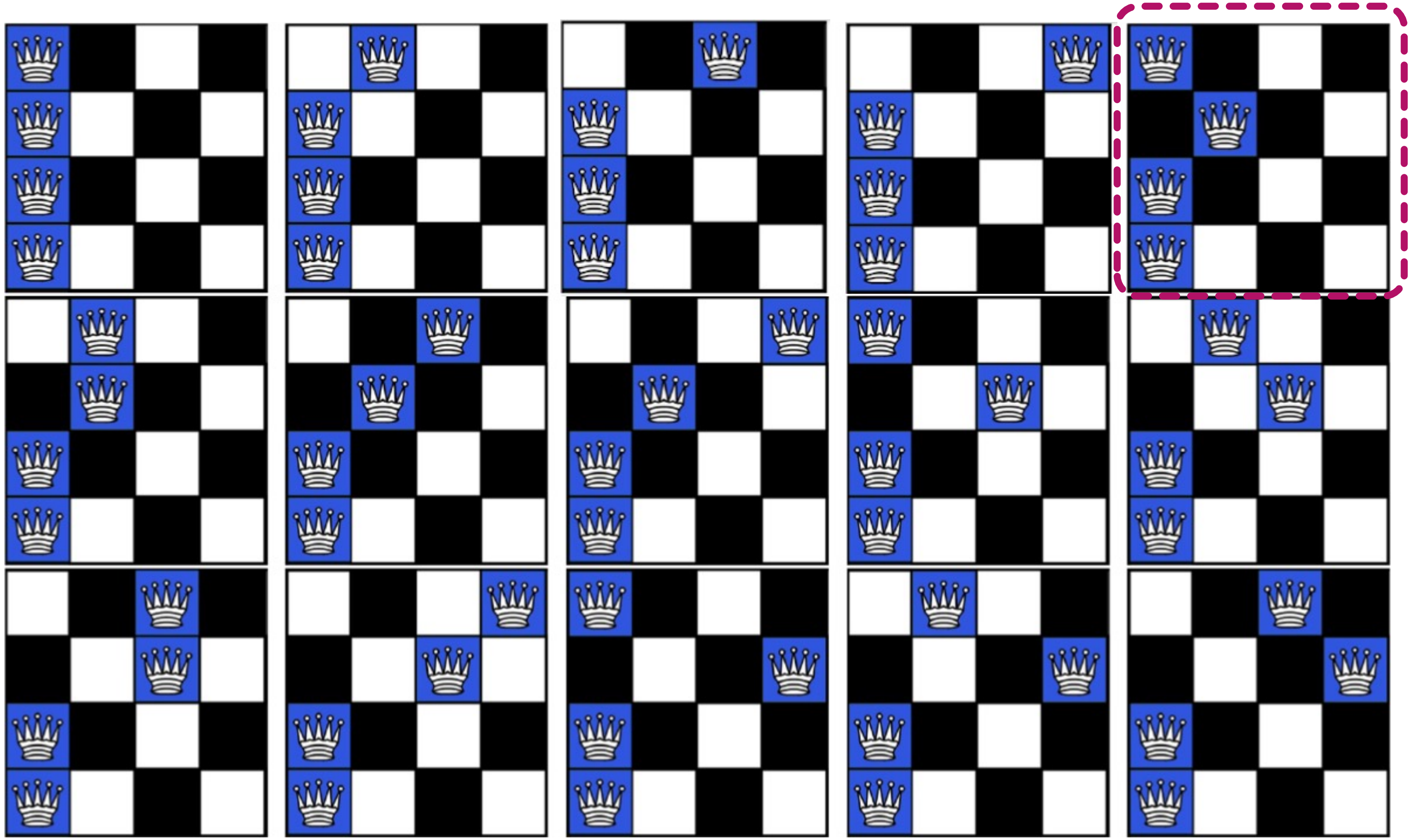
```
function BACKTRACKING-SEARCH(csp) returns solution/failure
    return RECURSIVE-BACKTRACKING({ }, csp)

function RECURSIVE-BACKTRACKING(assignment, csp) returns soln/failure
    if assignment is complete then return assignment
    var ← SELECT-UNASSIGNED-VARIABLE(VARIABLES[csp], assignment, csp)
    for each value in ORDER-DOMAIN-VALUES(var, assignment, csp) do
        if value is consistent with assignment given CONSTRAINTS[csp] then
            add $\{var = value\}$ to assignment
            result ← RECURSIVE-BACKTRACKING(assignment, csp)
            if $result \neq failure$ then return result
            remove $\{var = value\}$ from assignment
    return failure
```

Backtracking Search(BTS) for N-Queen



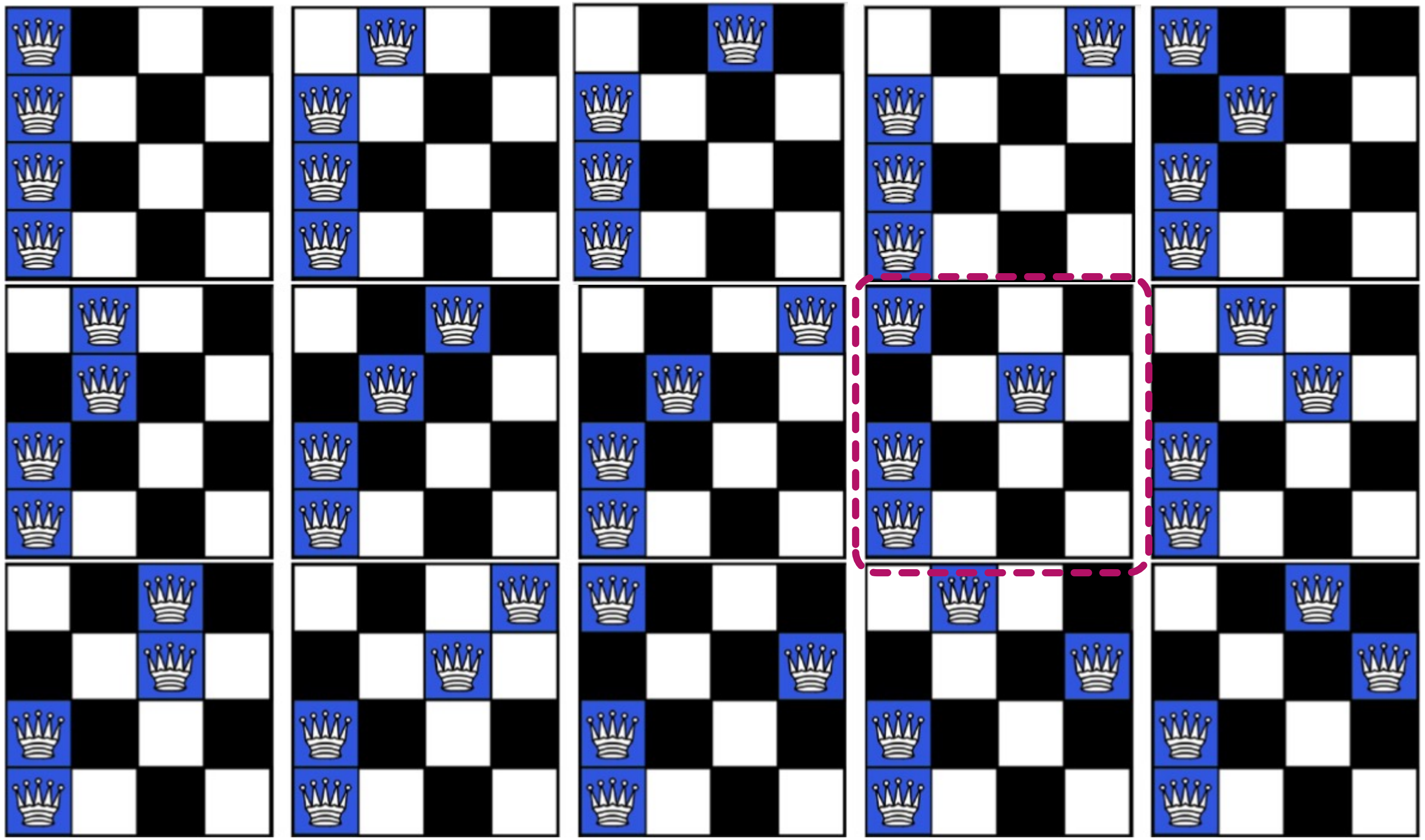
$N = 4$

Backtracking Search(BTS) for N-Queen



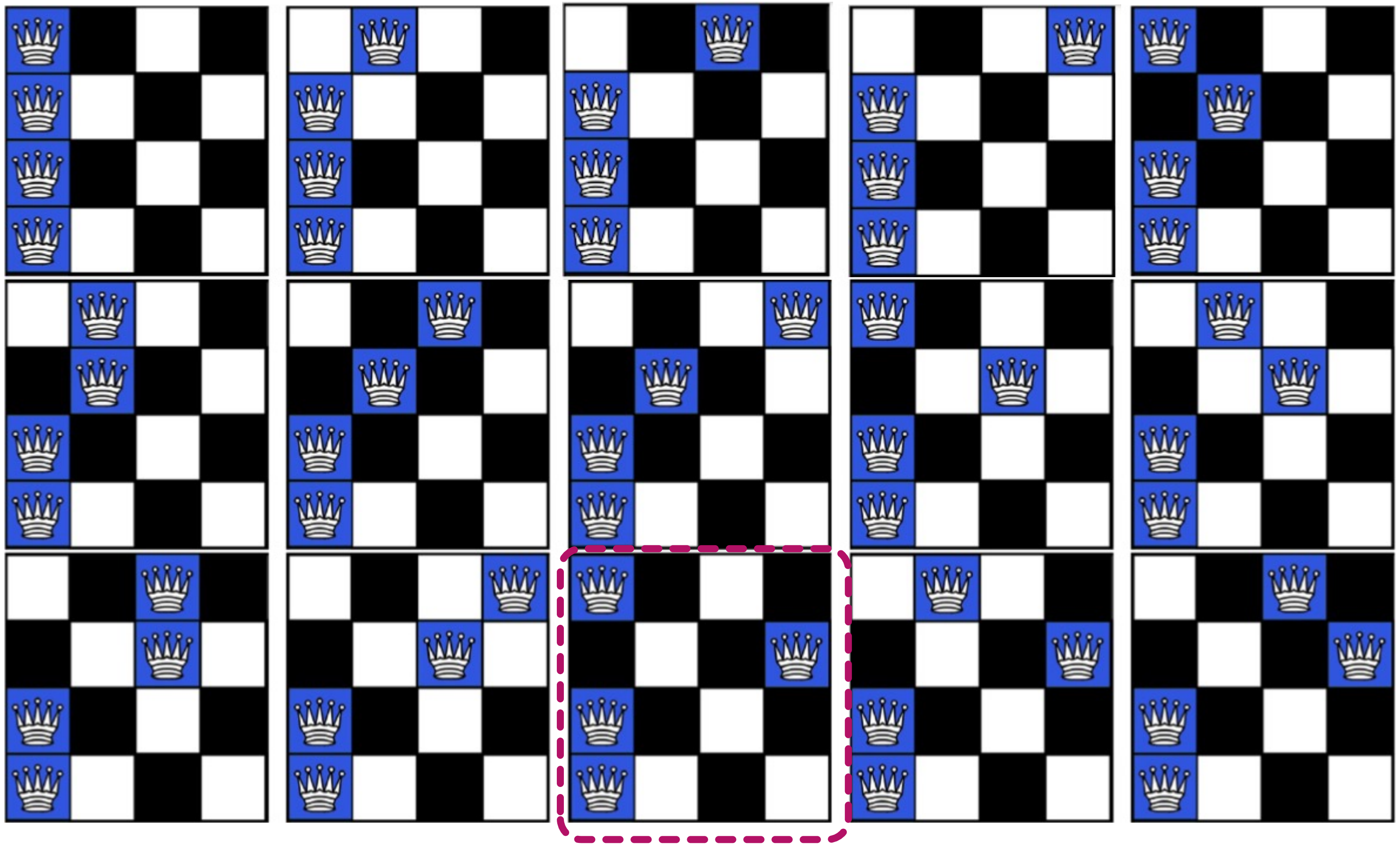
$N = 4$

Backtracking Search(BTS) for N-Queen



$N = 4$

Backtracking Search(BTS) for N-Queen



$N = 4$

continue...

Outline

- Problem description for the N-Queen problem
- Backtracking Search (BTS) for the N-Queen problem
- **Improving BTS for the N-Queen problem**
- Summary

How to improve BTS

- **Which variable should be assigned next?**
- **In what order should its values be tried?**
- **Can we detect inevitable failure early?**

Improve Backtracking Efficiency

- **Which variable should be assigned next?**

  Minimum Remaining Value: Choose the variable with the fewest legal values in its domain.

- **In what order should its values be tried?**

  Least Constraining Value: Given a variable, choose the least constraining value, i.e., the one that rules out the fewest values in the remaining variables

- **Can we detect inevitable failure early?**

  Inference: Forward checking and using constraint propagation, e.g., arc consistency test.

An Example for solving N-Queen

- **Minimum Remaining Value:**

Choose the variable with the fewest legal values in its domain.

- **Inference**:

forward checking, keep track of remaining legal values for the unassigned variables. Terminate when any variable has no legal values.

An Example for solving 8-Queen

Q1	Q	X	X	X	X	X	X	X	
Q2	X	X							6
Q3	X		X						6
Q4	X			X					6
Q5	X				X				6
Q6	X					X			6
Q7	X						X		6
Q8	X							X	6

- Start: All queens have 8 possible choices
- choose **Q1**, let Q1 = 1
- remove all assignments inconsistent with Q1 = 1

Q1	Q	X	X	X	X	X	X	X	
Q2	X	X	Q	X	X	X	X	X	
Q3	X	X	X	X					4
Q4	X		X	X	X				4
Q5	X		X		X	X			4
Q6	X		X			X	X		4
Q7	X		X				X	X	4
Q8	X		X					X	5

- All queens have 6 possible choices
- choose **Q2**, let Q2 = 3
- Continue to remove all assignments inconsistent with Q2 = 3
- Then: Q3:4 Q4:4 Q5:4 Q6:4 Q7:4 Q8:5

An Example for solving 8-Queen

Q1	Q	X	X	X	X	X	X	X	
Q2	X	X	Q	X	X	X	X	X	
Q3	X	X	X	X	Q	X	X	X	
Q4	X		X	X	X	X			3
Q5	X		X		X	X	X		3
Q6	X	X	X		X	X	X	X	1
Q7	X		X		X		X	X	3
Q8	X		X		X			X	4

Q1	Q	X	X	X	X	X	X	X	
Q2	X	X	Q	X	X	X	X	X	
Q3	X	X	X	X	Q	X	X	X	
Q4	X	X	X	X	X	X			2
Q5	X		X	X	X	X	X		2
Q6	X	X	X	Q	X	X	X	X	
Q7	X		X	X	X		X	X	2
Q8	X	X	X	X	X	X		X	1

- choose **Q3**, let Q3 = 5
- Continue to remove all assignments inconsistent with Q3 = 5
- Then: Q4:3 Q5:3 Q6:1 Q7:3 Q8:4

- choose **Q6**, let Q6 = 4
- Continue to remove all assignments inconsistent with Q6 = 4
- Then: Q4:2 Q5:2 Q7:2 Q8:1

An Example for solving 8-Queen

Q1	Q	X	X	X	X	X	X	X	
Q2	X	X	Q	X	X	X	X	X	
Q3	X	X	X	X	Q	X	X	X	
Q4	X	X	X	X	X	X			2
Q5	X		X	X	X	X	X		2
Q6	X	X	X	Q	X	X	X	X	
Q7	X		X	X	X	X	X	X	1
Q8	X	X	X	X	X	X	Q	X	

- choose **Q8**, let Q8 = 7
- Continue to remove all assignments inconsistent with Q8 = 7
- Then: Q4:2 Q5:2 Q7:1

Q1	Q	X	X	X	X	X	X	X	
Q2	X	X	Q	X	X	X	X	X	
Q3	X	X	X	X	Q	X	X	X	
Q4	X	X	X	X	X	X			2
Q5	X	X	X	X	X	X	X		1
Q6	X	X	X	Q	X	X	X	X	
Q7	X	Q	X	X	X	X	X	X	
Q8	X	X	X	X	X	X	Q	X	

- choose **Q7**, let Q7 = 2
- Continue to remove all assignments inconsistent with Q7 = 2
- Then: Q4:2 Q5:1

An Example for solving 8-Queen

Q1	Q	X	X	X	X	X	X	X	
Q2	X	X	Q	X	X	X	X	X	
Q3	X	X	X	X	Q	X	X	X	
Q4	X	X	X	X	X	X	X	X	0
Q5	X	X	X	X	X	X	X	Q	
Q6	X	X	X	Q	X	X	X	X	
Q7	X	Q	X	X	X	X	X	X	
Q8	X	X	X	X	X	X	Q	X	

- choose **Q5**, let Q5 = 8
- Continue to remove all assignments inconsistent with Q5 = 8
- **Then Q4 has no legal values, go back.**

Q1	Q	X	X	X	X	X	X	X
Q2	X	X	Q	X	X	X	X	X
Q3	X	X	X	X		Q		
Q4	X		X	X	X			
Q5	X		X		X	X		
Q6	X		X			X	X	
Q7	X		X				X	X
Q8	X		X					X

- go back to **Q3**, let Q3=6
- go on with the procedure
- …

Outline

- Problem description for the N-Queen problem
- Backtracking Search (BTS) for the N-Queen problem
- Improving BTS for the N-Queen problem
- **Summary**

Summary

- Formulate the N-Queen problem as a CSP
- Solve N-Queen with BTS
- Solve N-Queen with improving BTS

To do

- Practice 6: Solve N-Queen with BTS, DDL:
- Practice 7: Solve N-Queen with improving BTS, DDL: