Programming Language Theory
ICS313

Nancy E. Reed
nreed@hawaii.edu
Program Complexity

- Example problems
  - N-queens
  - cryptarithmatic

- Solution strategies
  - naïve (exponential)
  - ......
  - better $O(n)$, $O(n \times \log(n))$, or $O(n^2)$
  - $O(1)$

- Solutions
  - Prolog
  - Lisp

<table>
<thead>
<tr>
<th>$O(1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O(\log(n))$</td>
</tr>
<tr>
<td>$O(n)$</td>
</tr>
<tr>
<td>$O(n \times \log(n))$</td>
</tr>
<tr>
<td>$O(n^2)$</td>
</tr>
<tr>
<td>$O(n^3)$</td>
</tr>
<tr>
<td>$O(2^n)$</td>
</tr>
<tr>
<td>$O(n!)$</td>
</tr>
<tr>
<td>$O(n^n)$</td>
</tr>
</tbody>
</table>
Eight Queens Problem

- On an 8-by-8 chessboard, **place 8 queens** so that they do not threaten each other. Queens are the most powerful piece and can move any number of squares in any direction along a row, column, or a diagonal.

- Generalize to N queens problem where \( N \geq 4 \) (why won’t it work for \( N < 4 \)?)
Problem Complexity

- Naïve solution – 8 queens, each in one of 64 squares = \(64^8 = 281,474,976,710,656\) (over 281 trillion) (exponential/exponential)

- But, queens can’t be in the same square so there are \(\text{8 choose 64} = \frac{64!}{56! \times 8!} = 4,426,165,368\) (now only over 4 billion!) (factorial)

- But, the row numbers are a permutation of 1-8, and column numbers are the same, this helps a little, \(8! \times 8! = 40,320 \times 40,320 = 1,625,702,400\) (factorial)
Problem Complexity (cont)

- However, we can let $Q_n$ be the queen in column $n$, then the only thing left is to pick the row for each queen.

- The number of permutations of the integers 1 through 8 is $8! = 40,320$

This is down greatly from over 4 trillion, but this is still factorial in the number of queens.
How to Tell if Queens are on the Same Diagonal?

- Two queens are on the same upper left to bottom right diagonal iff the sum of the row and the column is the same for each.
- Two queens are on the same upper right to bottom left diagonal iff the difference between the row and the column are the same for each.

Prolog Solution to the 8 Queens problem

- [http://www2.hawaii.edu/~nreed/ics313/assignments/prolog/8queens.pro](http://www2.hawaii.edu/~nreed/ics313/assignments/prolog/8queens.pro)
Prolog Solution to 8 Queens

solve(P) :- permutation([1, 2, 3, 4, 5, 6, 7, 8], P),
            combine([1, 2, 3, 4, 5, 6, 7, 8], P, S, D),
            all_diff(S), % P is a solution,
            all_diff(D). % P is a permutation of
            % the numbers 1-8

permutation([], []).
permutation([X | Y], Z) :-
    permutation(Y, W), remove(X, Z, W).

remove(X, [X | R], R). % separate first part of list
remove(X, [F | R], [F | S]) :- remove(X, R, S).

combine([], [], [], []). % split lists
combine([X1 | X], [Y1 | Y], [S1 | S], [D1 | D]) :-
    S1 is X1 + Y1, D1 is X1 - Y1,
    combine(X, Y, S, D).

all_diff([X]).
all_diff([X | Y]) :- \=member(X, Y), all_diff(Y).
    % X is not a member of Y
Solving the N-Queens Problem Without Searching!

- Find one solution to the N-Queens problem for any $n \geq 4$
- Placing queens takes constant time/queen, $O(n)$
- Drawing the board takes $O(n^2)$ time

| $O(1)$ |
| $O(\log(n))$ |
| $O(n)$ |
| $O(n \times \log(n))$ |
| $O(n^2)$ |
| $O(n^3)$ |
| $O(2^n)$ |
| $O(n!)$ |
| $O(n^n)$ |
Lisp for the N-Queens Problem

- This polynomial time algorithm finds one solution!
- Note: printing the board is still \( n^2 \)
- A Lisp implementation is here: \texttt{N-Queens.lisp}
- For more, see Marty Hall’s pages
  - \url{http://www.apl.jhu.edu/~hall/NQueens.html}
Cryptarithmatic Problem

- crptarthur.pro
- Find the unique digit that will replace each character to solve this puzzle

```
SEND  
+MORE  
_____  
MONEY
```
Representing Numbers with functor digit

% digit is any single character 0-9

digit(0).
digit(1).
digit(2).
digit(3).
digit(4).
digit(5).
digit(6).
digit(7).
digit(8).
digit(9).

% carry digit must be 0 or 1

carry(0). carry(1).

% var = list of vars

DigitList = [S, E, N, D, M, O, R, Y],
Representing Addition

```
 1 2 3 4 5
```

```
SEND
+MORE
-------
MONEY
```

carry(M), M=1,  % because M is \textbf{carry} to col 1, M can’t be 0, therefore 1
digit(S), S\neq 0,  % S is not 0, because S is most significant digit in row
% S + M is at least 10 and M is 1, so S is 9 \{ or 8 \}

```
column(1, 0, 0, C2, M, 0, DigitList),  %[C_n are the carries
  column(2, S, M, C3, O, C2, DigitList),  % column 2, etc.
  column(3, E, O, C4, N, C3, DigitList),
  column(4, N, R, C4, E, C5, DigitList),
  column(5, D, E, 0, Y, C5, DigitList).
```