Homework #6

Surabhi Raghavan

October 15, 2024

Problem 4

Write an algorithm that takes an integer n as input and determines the total number of solutions to the n-Queens problem.

Solution

The n-queens problem involves placing n queens on a n x n chessboard such that no two queens threaten each other.
Two queens threaten each other if:

- if 2 queens are in the same row
- if 2 queens are in the same column
- if 2 queens are iin the same diagonal

We need two separate functions- one to place the queen in a particular position and another to determine all combinations of the solutions.

Alogrithm: Determine if a queen can be placed at a particular position
Input: k (int)- represents the kth queen
Output: Returns 1 if the queen can be placed and 0 if it cannot

Algorithm 1 Place(k, i)

```
for $j = 1$ to $k - 1$ do
    if $x[j] = i$ or $|j - k| = |x[j] - i|$ then
        return 0
    end if
end for
return 1
```

Alogrithm: Function to determine the number of solutions to the n-queens problem
Input: n (int)- represents the number of queens and the size of the chessboard
Output: Number of solutions

Algorithm 2 NQueens(n)

```
$k \leftarrow 1$
$x[k] \leftarrow 0$
$solution\_count \leftarrow 0$                    ▷ Initialize counter for number of solutions
while $k > 0$ do
    $x[k] \leftarrow x[k] + 1$
    while $x[k] \leq n$ and not PLACE$(k, x[k])$ do
        $x[k] \leftarrow x[k] + 1$
    end while
    if $x[k] \leq n$ then
        if $k == n$ then
            $solution\_count \leftarrow solution\_count + 1$      ▷ Increment solution count
        else
            $k \leftarrow k + 1$
            $x[k] \leftarrow 0$
        end if
    else
        $k \leftarrow k - 1$
    end if
end while
return $solution\_count$                      ▷ Return the total number of solutions found
```

Problem 10

Find at least two instances of the n-Queens problem that have no solutions.

Solution

There are 2 cases in which the n-Queens problem has no solutions.
I figured it out by manually placing queens on the chessboard. It was quickly determined that there was no position where the queens didn't threaten each other.

CASE 1: No Solution for $n = 2$

Consider a 2 x 2 chessboard. When $n = 2$, we need to place 2 queens on the chessboard, with each queen occupying its own row and column. We start by placing the first queen at position a2 (as shown in Fig. 1). Now, we need to place the second queen in one of the remaining three positions. However, no matter where we place the second queen, it will either be in the same row, the same column, or the same diagonal as the first queen.

- The queen cannot be placed at b2, as it would be in the same row as the first queen (Fig. 2).

- The queen cannot be placed at a1, as it would be in the same column as the first queen (Fig. 3).
- The queen cannot be placed at b1, as it would be in the same diagonal as the first queen (Fig. 4).

Thus, there is no valid way to place two queens on a 2x2 chessboard without them threatening each other.

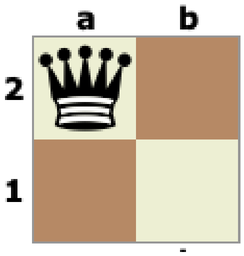

Figure 1: Initial Positioning of the Queen (a2)

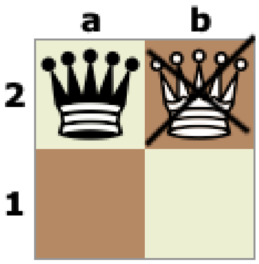

Figure 2: Queen cannot be placed in the same row (b2)

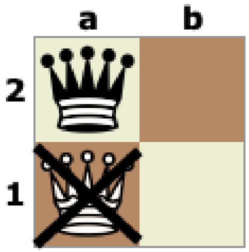

Figure 3: Queen cannot be placed in the same column (a1)

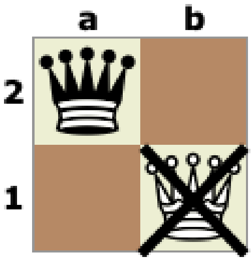

Figure 4: Queen cannot be placed on the diagonal (b1)

CASE 2: No Solution for $n = 3$

Consider a 3 x 3 chessboard. When $n = 3$, we need to place 3 queens on the chessboard, with each queen occupying its own row and column. We start by placing the first queen at position a3 (as shown in Fig. 1). Now, we need to place the second queen in one of the remaining three positions. However, no matter where we place the second queen, it will either be in the same row, the same column, or the same diagonal as the first queen.

- The queen cannot be placed at b3 or c3, as it would be in the same row as the first queen (Fig. 6).

- The queen cannot be placed at a2 or a1, as it would be in the same column as the first queen (Fig. 7).
- The queen cannot be placed at b2 or c1, as it would be in the same diagonal as the first queen (Fig. 8).

The only positions remaining are c2 or b1. We can place the second queen in either of these positions. I will choose to place it at c2 (Fig. 9). Fig 10: Shows us that there is no other position to place the third queen. The same would be shown for any other position of the queens.

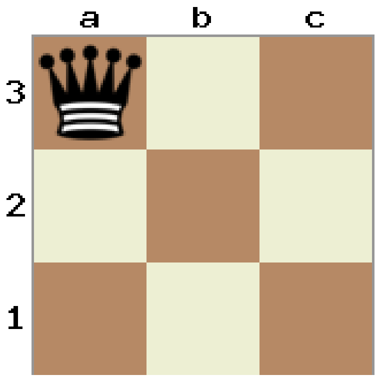

Figure 5: Queen at a3

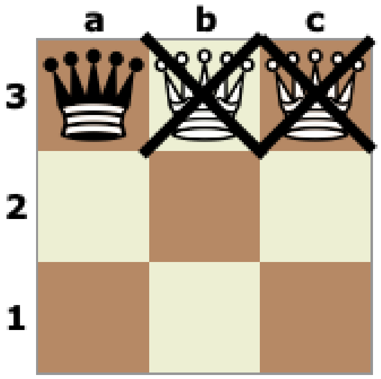

Figure 6: Cannot be placed in the same row (b3, c3)

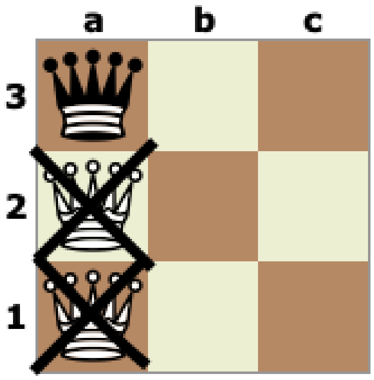

Figure 7: Cannot be placed in the same column (a2, a1)

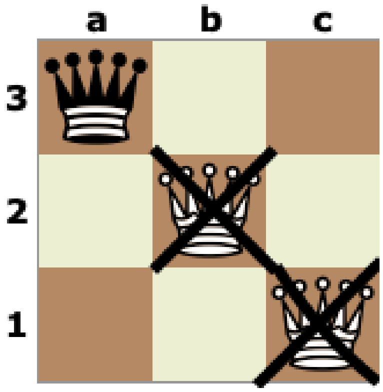

Figure 8: Cannot be placed on the diagonal (b2, c1)

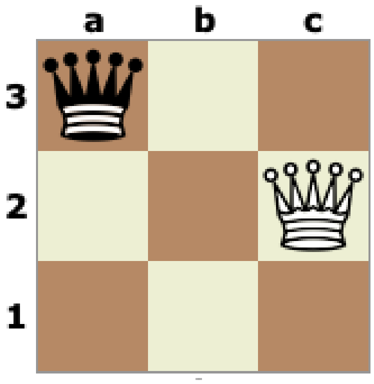

Figure 9: Second queen placed at c2

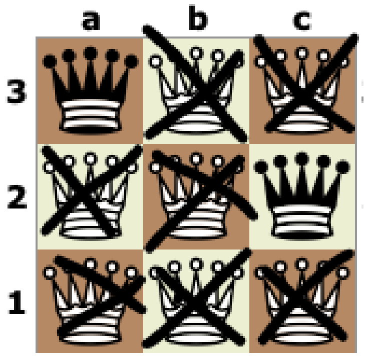

Figure 10: Third queen scenario- cannot be placed anywhere

Problem 13

Use the Backtracking algorithm for the Sum-of-Subsets problem (Algorithm 5.4) to find all combinations of the following numbers that sum to $W = 52$:

$$w_1 = 2, \quad w_2 = 10, \quad w_3 = 13, \quad w_4 = 17, \quad w_5 = 22, \quad w_6 = 42$$

Show the actions step by step.

Solution

Algorithm 3 sum_of_subsets(index i, int weight, int total)

```
if promising($i$) then
    if weight = $W$ then
        Output: subset from $include[1]$ through $include[i]$
    else
        include[$i + 1$] = ”yes”
        sum_of_subsets($i + 1$, weight + $w[i + 1]$, total - $w[i + 1]$)    ▷ Include $w[i + 1]$
        include[$i + 1$] = ”no”
        sum_of_subsets($i + 1$, weight, total - $w[i + 1]$)    ▷ Do not include $w[i + 1]$
    end if
end if
```

Promising Function

Algorithm 4 promising(index i)

```
Return: $(weight + total \geq W)$ and $(weight + w[i + 1] \leq W)$
```

Initial Setup

- **Target Sum (W):** 52
- **Weights (w):** { 2, 10, 13, 17, 22, 42 }
- **Include Array:** Initially empty, i.e., { }
- **Current Weight (W):** Initially 0

The **Promising Function** checks two conditions:

1. $\text{curr\_sum} + \text{remaining\_weights} \geq W$ — Ensures that we can still reach the target sum with the remaining weights.
2. $\text{curr\_sum} + \text{next\_weight} \leq W$ — Ensures that adding the next weight will not exceed the target sum.

Step-by-Step Backtracking Process with Promising Function Explanation

Step 1:

Action: Promising function returns true, so we include weight 2.
Current State:

- Include array: { 2 }
- Current Weight (W): 2

Explanation of Promising Function:

$$2 + (10 + 13 + 17 + 22 + 42) = 106 \geq 52 \quad \text{(we can still reach the target sum)}$$

$$2 + 10 = 12 \leq 52 \quad \text{(adding the next weight won't exceed the target)}$$

Step 2:

Action: Promising function returns true, so we include weight 10.
Current State:

- Include array: { 2, 10 }
- Current Weight (W): 12

Explanation of Promising Function:

$$12 + (13 + 17 + 22 + 42) = 104 \geq 52 \quad \text{(we can still reach the target sum)}$$

$$12 + 13 = 25 \leq 52 \quad \text{(adding the next weight won't exceed the target)}$$

Step 3:

Action: Promising function returns true, so we include weight 13.
Current State:

- Include array: { 2, 10, 13 }
- Current Weight (W): 25

Explanation of Promising Function:

$$25 + (17 + 22 + 42) = 81 \geq 52 \quad \text{(we can still reach the target sum)}$$

$$25 + 17 = 42 \leq 52 \quad \text{(adding the next weight won't exceed the target)}$$

Step 4:

Action: Promising function returns true, so we include weight 17.
Current State:

- Include array: { 2, 10, 13, 17 }
- Current Weight (W): 42

Explanation of Promising Function:

$$42 + (22 + 42) = 64 \geq 52 \quad \text{(we can still reach the target sum)}$$

$$42 + 22 = 64 > 52 \quad \text{(adding the next weight will exceed the target)}$$

Step 5:

Action: Backtracking begins. Exclude the last included weight 17.
Current State:

- Include array: { 2, 10, 13 }
- Current Weight (W): 25

Explanation: The promising function returns false for weight 17 since adding more exceeds 52. Backtrack and try the next weight.

Step 6:

Action: Promising function returns true, so we include weight 22.
Current State:

- Include array: { 2, 10, 13, 22 }
- Current Weight (W): 47

Explanation of Promising Function:

$$25 + (22 + 42) = 64 \geq 52 \quad \text{(we can still reach the target sum)}$$

$$25 + 22 = 47 \leq 52 \quad \text{(adding the next weight won't exceed the target)}$$

Step 7:

Action: Backtracking occurs. Exclude the last included weight 22.
Current State:

- Include array: { 2, 10, 13 }
- Current Weight (W): 25

Explanation: The promising function returns false for weight 22 because adding more weights will exceed 52. Backtrack and try other combinations.

Step 8:

Action: Backtracking occurs. Exclude the last included weight 13.
Current State:

- Include array: { 2, 10 }
- Current Weight (W): 12

Explanation: The promising function returns false for the subset { 2, 10, 13 } since adding more exceeds 52. Backtrack to explore more possibilities.

Step 9:

Action: Promising function returns true, so we include weight 17.
Current State:

- Include array: { 2, 10, 17 }
- Current Weight (W): 29

Explanation of Promising Function:

$$12 + (17 + 22 + 42) = 106 \geq 52 \quad \text{(we can still reach the target sum)}$$

$$29 + 22 = 51 \leq 52 \quad \text{(adding the next weight won't exceed the target)}$$

Step 10:

Action: Promising function returns true, so we include weight 22.
Current State:

- Include array: { 2, 10, 17, 22 }
- Current Weight (W): 51

Explanation of Promising Function:

$$29 + (22 + 42) = 93 \geq 52 \quad \text{(we can still reach the target sum)}$$

$$51 + 42 = 93 > 52 \quad \text{(adding the next weight will exceed the target)}$$

Step 11:

Action: Backtracking occurs. Exclude the last included weight 22.
Current State:

- Include array: { 2, 10, 17 }
- Current Weight (W): 29

Explanation: The promising function returns false when attempting to include more weights beyond 51. Backtrack to explore further combinations.

Step 12:

Action: Promising function returns true, so we include weight 22.
Current State:

- Include array: { 2, 22 }
- Current Weight (W): 24

Explanation of Promising Function:

$$2 + (22 + 42) = 66 \geq 52 \quad \text{(we can still reach the target sum)}$$

$$24 + 42 = 66 > 52 \quad \text{(adding the next weight will exceed the target)}$$

Step 13:

Action: Promising function returns true, so we include weight 42.
Current State:

- Include array: { 2, 42 }
- Current Weight (W): 44

Explanation of Promising Function:

$$24 + 42 = 66 > 52 \quad \text{(adding the next weight will exceed the target)}$$

Step 14:

Action: Backtracking occurs. Exclude the last included weight 42.
Current State:

- Include array: { 2 }
- Current Weight (W): 2

Explanation: The promising function returns false for { 2, 42 } since it exceeds 52. Backtrack and try other combinations.

Step 15:

Action: Promising function returns true, so we include weight 10.
Current State:

- Include array: { 10 }
- Current Weight (W): 10

Explanation of Promising Function:

$$10 + (13 + 17 + 22 + 42) = 104 \geq 52 \quad \text{(we can still reach the target sum)}$$

$$10 + 13 = 23 \leq 52 \quad \text{(adding the next weight won't exceed the target)}$$

Step 16:

Action: Promising function returns true, so we include weight 42.
Current State:

- Include array: { 10, 42 }
- Current Weight (W): 52

Explanation of Promising Function:

$$10 + (42) = 52 \quad \text{(exactly equals the target sum)}$$

Step 17:

Action: Backtracking occurs. Exclude weight 42.
Current State:

- Include array: { 10 }
- Current Weight (W): 10

Explanation: The promising function returned true for the subset { 10, 42 }, so now backtrack to explore further possibilities.

Step 18:

Action: Promising function returns true, so we include weight 13.
Current State:

- Include array: { 13 }
- Current Weight (W): 13

Explanation of Promising Function:

$$13 + (17 + 22 + 42) = 94 \geq 52 \quad \text{(we can still reach the target sum)}$$

$$13 + 17 = 30 \leq 52 \quad \text{(adding the next weight won't exceed the target)}$$

Step 19:

Action: Promising function returns true, so we include weight 17.
Current State:

- Include array: { 13, 17 }
- Current Weight (W): 30

Explanation of Promising Function:

$$30 + (22 + 42) = 94 \geq 52 \quad \text{(we can still reach the target sum)}$$

$$30 + 22 = 52 \quad \text{(adding the next weight exactly equals the target)}$$

Step 20:

Action: Promising function returns true, so we include weight 22.
Current State:

- Include array: { 13, 17, 22 }
- Current Weight (W): 52

Explanation of Promising Function:

$$30 + (22) = 52 \quad \text{(exactly equals the target sum)}$$

Conclusion

- **Solution 1:** { 10, 42 }
- **Solution 2:** { 13, 17, 22 }