

**Shri Vile Parle Kelavani Mandal’s Institute of Technology, Dhule**

**Department of Information Technology**

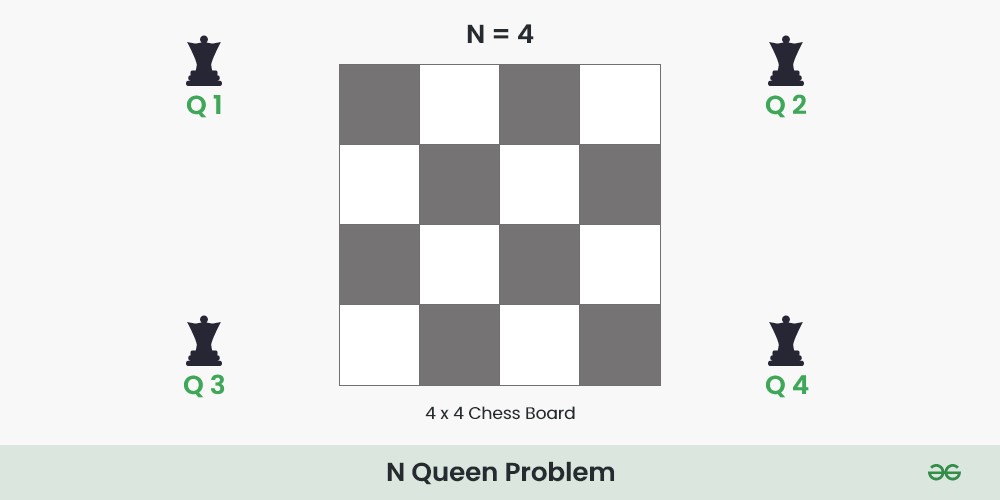
**Design and Analysis of Algorithm Lab**

|  |  |  |  |
| --- | --- | --- | --- |
| **NAME** | **ROLL NO** | **PRN** | **CLASS** |
| PAVANRAJ RAVINDRA PATIL | 43 | 2254491246044 | SY(IT) |
| PRAFULL DILIP PATIL | 44 | 2254491246045 | SY(IT) |
| PRANIT RAJENDRA PATIL | 45 | 2254491246046 | SY(IT) |
| ROHIT RAJENDRA PATIL | 46 | 2254491246047 | SY(IT) |

Q.1. How 4-Queen problem is solved by backtracking approach? Explain with the help of state space tree.

The **4 Queens** Problem consists in placing four queens on a **4 x 4** chessboard so that no two queens attack each other. That is, no two queens are allowed to be placed on the **same row**, the **same column** or the**same diagonal**.

We are going to look for the solution for n=4 on a 4 x 4 chessboard in this article.



## 4 Queens Problem using Backtraking Aglorithm:

Place each queen one by one in different rows, starting from the topmost row. While placing a queen in a row, check for clashes with already placed queens. For any column, if there is no clash then mark this row and column as part of the solution by placing the queen. In case, if no safe cell found due to clashes, then backtrack (i.e, undo the placement of recent queen) and return false*.*

### Illustration of 4 Queens Solution:-

**Step 1:** Initialize a 4×4 board.

0 1 2 3

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

0

1

2

3

**Step 2:**

* Put our first Queen (**Q1**) in the**(0,0)** cell .
* ‘**x**‘ represents the cells which is not safe i.e. they are under attack by the Queen (**Q1**).
* After this move to the next row [ 0 -> 1 ].

0 1 2 3

|  |  |  |  |
| --- | --- | --- | --- |
| Q1 | X | X | X |
| X | X |  |  |
| X |  | X |  |
| X |  |  | X |

0

1

2

3

**Step 3:**

* Put our next Queen (**Q2**) in the **(1,2)**cell .
* After this move to the next row [ 1 -> 2 ].

0 1 2 3

|  |  |  |  |
| --- | --- | --- | --- |
| Q1 | X | X | X |
| X | X | Q2 | X |
| X | X | X | X |
| X |  | X | X |

0

1

2

3

**Step 4:**

* At row 2 there is no cell which are safe to place Queen (**Q3**) .
* So, backtrack and remove queen **Q2** queen from cell ( 1, 2 ) .

**Step 5:**

* There is still a safe cell in the row 1 i.e. cell ( 1, 3 ).
* Put Queen ( **Q2** ) at cell ( 1, 3).

0 1 2 3

|  |  |  |  |
| --- | --- | --- | --- |
| Q1 | X | X | X |
| X | X | X | Q2 |
| X |  | X | X |
| X | X |  | X |

0

1

2

3

**Step 6:**

* Put queen**( Q3 )** at cell ( 2, 1 ).

|  |  |  |  |
| --- | --- | --- | --- |
| Q1 | X | X | X |
| X | X | X | Q2 |
| X | Q3 | X | X |
| X | X | X | X |

0 1 2 3

0

1

2

3

**Step 7:**

* There is no any cell to place Queen ( **Q4**) at row 3.
* Backtrack and remove Queen ( **Q3 )** from row 2.
* Again there is no other safe cell in row 2, So backtrack again and remove queen (**Q2** ) from row 1.
* Queen ( **Q1 )** will be remove from cell **(0,0)**and move to next safe cell i.e. **(0 , 1)**.

**Step 8:**

* Place Queen Q1 at cell (0 , 1), and move to next row.

0 1 2 3

|  |  |  |  |
| --- | --- | --- | --- |
| X | Q1 | X | X |
| X | X | X | Q2 |
|  | X | X | X |
|  | X |  | X |

0

1

2

3

**Step 9:**

* Place Queen **Q3** at cell **(2 , 0)**, and move to next row.

0 1 2 3

|  |  |  |  |
| --- | --- | --- | --- |
| X | Q1 | X | X |
| X | X | X | Q2 |
| Q3 | X | X | X |
| X | X |  | X |

0

1

2

3

**Step 10:**

* Place Queen **Q4** at cell **(3 , 2)**, and move to next row.
* This is one possible configuration of solution.

0 1 2 3

|  |  |  |  |
| --- | --- | --- | --- |
| X | Q1 | X | X |
| X | X | X | Q2 |
| Q3 | X | X | X |
| X | X | Q4 | X |

0

1

2

3

Similarly, find second possible configuration of solution

0 1 2 3

|  |  |  |  |
| --- | --- | --- | --- |
| X | X | Q1 | X |
| Q2 | X | X | X |
| X | X | X | Q3 |
| X | Q4 | X | X |

0

1

2

3