Backtracking

- Some problems can be solved, by exhaustive search.
- Backtracking is a more intelligent variation of this approach.
- The principal idea is to construct solutions one component at a time and evaluate such partially constructed candidates as follows.
  - If a partially constructed solution can be developed, continue.
  - If there is no legitimate option for the next component, **backtrack** to replace the last component of the partially constructed solution with its next option.

Backtracking

- **State-space tree,** represents the processing
- Its root represents an initial state
- The nodes of the first level in the tree represent the choices made for the first component of a solution
- The nodes of the second level represent the choices for the second component, and so on.
- A node in a state-space tree is said to be promising
  - if it corresponds to a partially constructed solution that may still lead to a complete solution;
  - otherwise, it is called nonpromising.
- Leaves represent either nonpromising dead ends or complete solutions found by the algorithm.

Backtracking

- In the majority of cases, a states-pace tree for a backtracking algorithm is constructed in the manner of depth-first search.
- If the algorithm reaches a complete solution to the problem, it either stops (if just one solution is required) or continues searching for other possible solutions.

N-Queens Problem

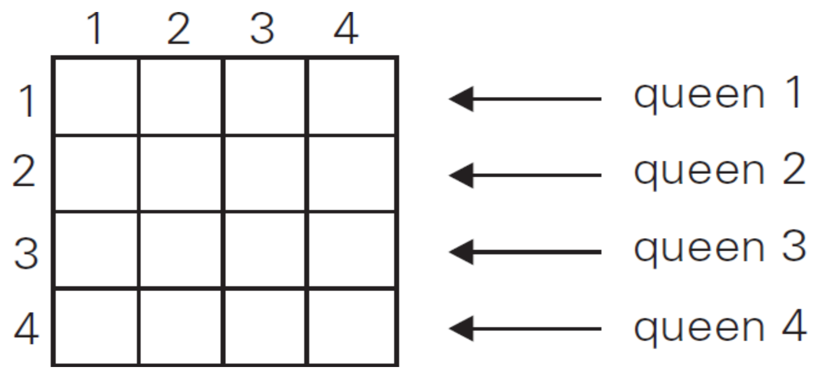
Problem Definition

- The problem is to place n queens on an **n × n** chessboard so that no two queens attack each other by being in the same row or in the same column or on the same diagonal.

- So let us consider the **4-queens problem** and solve it by the backtracking technique.

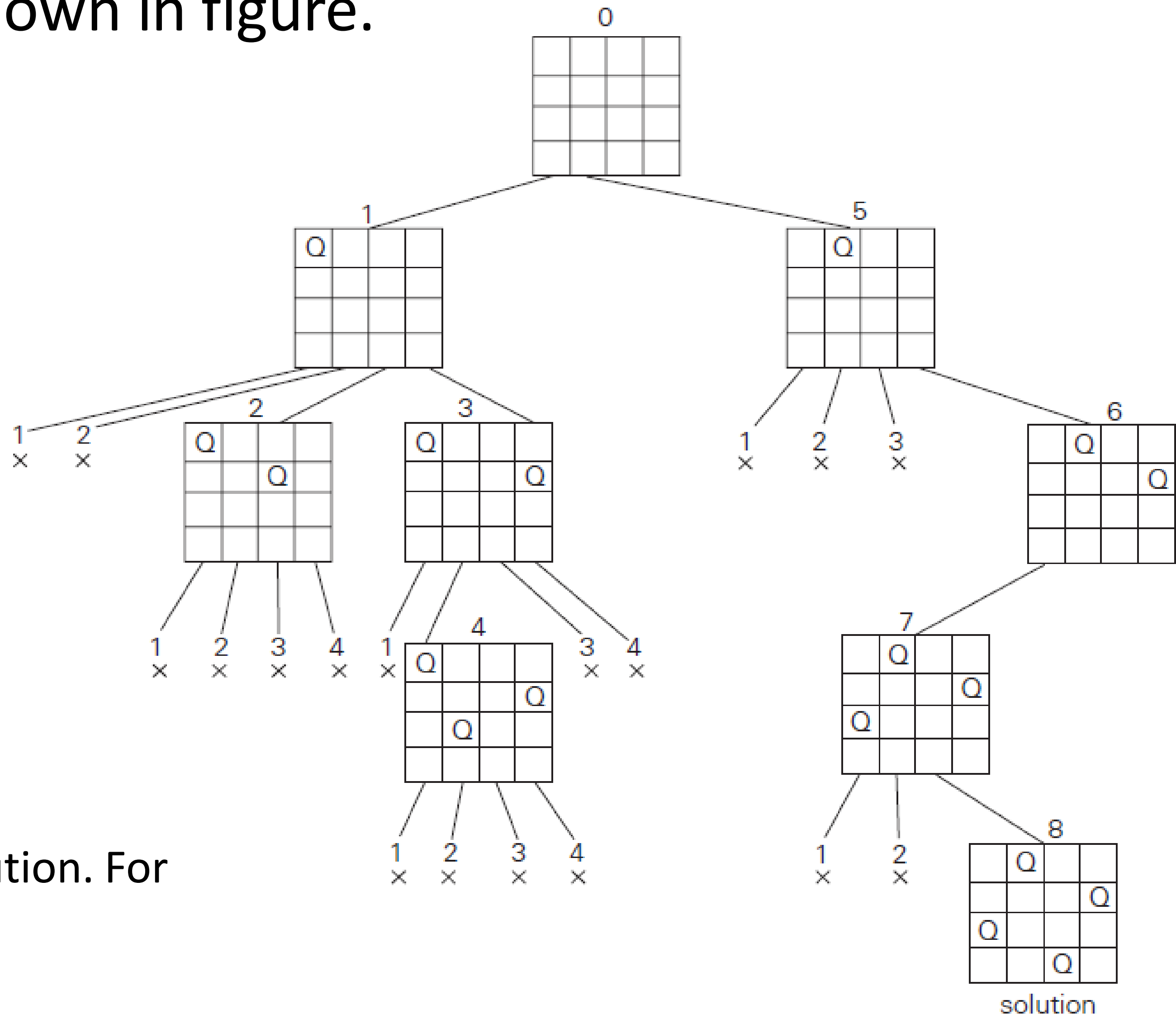
4-Queens problem

- Since each of the four queens has to be placed in its own row, all we need to do is to assign a column for each queen on the board presented in figure.



4-Queens problem

- The state-space tree of this search is shown in figure.



Note: This tree shows expansion of nodes till it gets a solution. For multiple solutions it need to be expanded further

4-queens problem

- If other solutions need to be found, the algorithm can simply resume its operations at the leaf at which it stopped.
- Alternatively, we can use the board’s symmetry for this purpose.
- Finally, it should be pointed out that a single solution to the n-queens problem for any $n \geq 4$ can be found in **linear time**.

N-queens prblem

Algorithm to find all solutions of n-queens problem

```
Algorithm NQueens(k, n)
// Using backtracking, this procedure prints all
// possible placements of n queens on an n × n
// chessboard so that they are nonattacking.
{
    for i := 1 to n do
    {
        if Place(k, i) then
        {
            x[k] := i;
            if (k = n) then write (x[1 : n]);
            else NQueens(k + 1, n);
        }
    }
}
```

N-queens prblem

Algorithm Place(k, i)
// Returns **true** if a queen can be placed in kth row and
// ith column. Otherwise it returns **false**. $x[\,]$ is a
// global array whose first $(k-1)$ values have been set.
// Abs(r) returns the absolute value of r.
{
    for $j := 1$ **to** $k-1$ **do**
        if (($x[j] = i$) // Two in the same column
            or (Abs($x[j] - i$) = Abs($j - k$)))
                // or in the same diagonal
            then return false;
    return true;
}

is_safe()

Applications

VLSI testing
Traffic control
Parallel memory storage scheme and deadlock prevention
Travelling sales person problem