4.3. Experiment No. 3

Aim:

Implement a solution for a Constraint Satisfaction Problem using Branch and Bound and Backtracking for n-queens problem or a graph coloring problem

Objective:

Implement a solution for a Constraint Satisfaction Problem using Branch and Bound and

Backtracking for n-queens problem or a graph coloring problem

Theory:

8 Queens Problem using Branch and Bound

The N-Queens problem is a puzzle of placing exactly N queens on an NxN chessboard, such that no two queens can attack each other in thatconfiguration. Thus, no two queens can lie in the same row, column ordiagonal.

The branch and bound solution is somehow different, it generates a partial solution until it figures that there's no point going deeperas we would ultimately lead to a dead end.

In the backtracking approach, we maintain an 8x8 binary matrix for keeping track of safe cells (by eliminating the unsafe cells, those that are likely to be attacked) and update it each time we place a new queen. However, it required O(n^2) time to check safe cell and update the queen.

In the 8 queens problem, we ensure the following:

1. no two queens share a row
2. no two queens share a column
3. no two queens share the same left diagonal
4. no two queens share the same right diagonal

ensure that the queens do not share the same column by theway we fill out our auxiliary matrix (column by column). Hence, onlythe left out 3 conditions are left out to be satisfied.

Applying the branch and bound approach:

The branch and bound approach suggest that we create a partial solution and use it to ascertain whether we need to continue in a particular direction or not. For this problem, we create 3 arrays to checkfor conditions 1,3 and 4. The Boolean arrays tell which rows and diagonals are already occupied. To achieve this, we need a numbering system to specify which queen is placed.

The indexes on these arrays would help us know which queen we are analysing.

Preprocessing - create two NxN matrices, one for top-left to bottom- right diagonal, and other for top-right to bottom-left diagonal. We need to fill these in such a way that two queens sharing same top- left bottom-right diagonal will have same value in slash Diagonal and two queens sharing same top-right bottom-left diagonal will have same value in back Slash Diagonal.

slashDiagnol(row)(col) = row + col

backSlashDiagnol(row)(col) = row - col

+ (N-1) { N = 8 }

{ we added (N-1) as we do not need negative values in backSlashDiagnol }

For placing a queen *i* on row *j*, check the following :

1. whether row 'j' is used or not
2. whether slashDiagnol 'i+j' is used or not
3. whether backSlashDiagnol 'i-j+7' is used or not

If the answer to any one of the following is true, we try another location for queen **i** on row **j**, mark the row and diagnols; and recur for queen **i+1**.

Applications:

Constraint satisfaction problem in AI has a wide range of applications, including scheduling, resource allocation, and automated reasoning.

Branch and bound is an algorithm used to solve combinatorial optimization problems

Input: n=8

Output- for (n = 8)

1	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	**0**	**0**	**0**	**0**	**0**	**1**	**0**
0	**0**	**0**	**0**	**1**	**0**	**0**	**0**
0	**0**	**0**	**0**	**0**	**0**	**0**	**1**
0	**1**	**0**	**0**	**0**	**0**	**0**	**0**
0	**0**	**0**	**1**	**0**	**0**	**0**	**0**
0	**0**	**0**	**0**	**0**	**1**	**0**	**0**
0	**0**	**1**	**0**	**0**	**0**	**0**	**0**

Graph coloring problem's solution using backtracking algorithmGraph coloring

The **graph coloring problem** is to discover whether the nodes of the graph G can be covered in such a way, that no two adjacent nodes have the same color yet only m colors are used. This graph coloring problem is also known as M- colorability decision problem.

The M – colorability optimization problem deals with the smallest integer m for which the graph G can be colored. The integer is known as a chromatic number of the graph.

Here, it can also be noticed that if d is the degree of the given graph, then it can be colored with d+ 1 color.

A graph is also known to be planar if and only if it can be drawn in a planar in such a way that no two edges cross each other. A special case is the 4 - colors problem for planar graphs. The problem is to color the region in a map in such away that no two adjacent regions have the same color. Yet only four colors are needed. This is a problem for which graphs are very useful because a map can beeasily transformed into a graph. Each region of the map becomes the node, and iftwo regions are adjacent, they are joined by an edge.

Graph coloring problem can also be solved using a state space tree, whereby applying a backtracking method required results are obtained.

For solving the **graph coloring problem**, we suppose that the graph

7	6	5	4	3	2	1	0
8	7	6	5	4	3	2	1
9	8	7	6	5	4	3	2
10	9	8	7	6	5	4	3
11	10	9	8	7	6	5	4
12	11	10	9	8	7	6	5
13	12	11	10	9	8	7	6
14	13	12	11	10	9	8	7

slash diagnol[row][col] = row + col

0	1	2	3	4	5	6	7
1	2	3	4	5	6	7	8
2	3	4	5	6	7	8	9
3	4	5	6	7	8	9	10
4	5	6	7	8	9	10	11
5	6	7	8	9	10	11	12
6	7	8	9	10	11	12	13
7	8	9	10	11	12	13	14

backslash diagnol[row][col] = row-col+(N-1)

is represented by its adjacency matrix G[1:n, 1:n], where, G[i, j]= 1 if (i, j) is an edge of G, and G[i, j] = 0 otherwise.

The colors are represented by the integers 1, 2, ..., m and the solutions are given by the n-tuple (x1, x2, x3, ..., xn), where x1 is the color of node i.

Algorithm for finding the m - colorings of

```
{
Repeat
{
// generate all legal assignments
for x[k], Next value (k); // assign
to x[k] a legal color.
 If ( x[k] = 0 ) then return; // no new color
 possible
 If (k = n) then // at most m colors have been
 used to color the n
vertices.
  Write (x[1 : n ]);
 Else mcoloring (k
 + 1);
}
 Until (false);
}
```

graph

Conclusion:

Thus we have implemented pre-processing of a text document such as stop word removal, stemming.

Outcome:

Upon completion of this experiment, students will be able to: Implement a solution for a Constraint Satisfaction Problem using Branch and Bound and Backtracking for n-queens problem or a graph coloring problem

Questions:

Q.1) Which are the constraints required to solve N Queen problem?

Q.2) Compare backtracking and branch and bound method

Q.3) What do mean by constraints satisfaction problem.