**Sudarshan Zunja**

***211IT072***

***DSA Assignment 5***

**Graph colouring** refers to the problem of colouring vertices of a graph in such a way that no two adjacent vertices have the same colour. This is also called the**vertex colouring** problem.

The backtracking algorithm makes the process of solving the Graph Colouring problem efficient by avoiding many bad decisions made in naïve approaches. In this approach, we colour a single vertex and then move to its adjacent (connected) vertex to colour it with a different colour. After colouring, we again move to another uncoloured adjacent vertex and repeat the process until all vertices of the given graph are coloured.

In case we find a vertex that has all adjacent vertices coloured and no colour is left to make it colour different, we backtrack and change the colour of the last-coloured vertices and again proceed further. If by backtracking, we come back to the same vertex from where we started and all colours were tried on it, then it means the given number of colours is insufficient to colour the given graph and we require more colours.

* This problem can be solved using backtracking algorithms as follows:
  + List down all the vertices and colours in two lists
  + Assign colour 1 to vertex 1
  + If vertex 2 is not adjacent to vertex 1 then assign the same colour, otherwise assign colour 2.
  + Repeat the process until all vertices are coloured.
* Algorithm backtracks whenever colour i is not possible to assign to any vertex k and it selects the next colour i + 1 and the test is repeated.

Test Case 0

var randomSudoku = [

  [11, 2, 0, 14, 3, 0, 0, 6, 7, 8, 0, 10, 0, 0, 15, 16],

  [0, 4, 0, 0, 2, 0, 0, 7, 0, 0, 16, 12, 0, 5, 14, 0],

  [0, 0, 0, 0, 10, 0, 0, 0, 1, 2, 0, 5, 0, 3, 7, 0],

  [0, 9, 7, 3, 8, 12, 0, 13, 6, 0, 14, 15, 1, 4, 10, 2],

  [0, 3, 2, 0, 5, 16, 8, 0, 0, 6, 10, 0, 0, 15, 12, 13],

  [6, 0, 8, 7, 1, 3, 0, 0, 12, 0, 15, 0, 2, 10, 0, 14],

  [13, 0, 12, 0, 9, 10, 0, 2, 4, 0, 3, 0, 16, 1, 5, 6],

  [9, 10, 14, 16, 0, 6, 12, 11, 5, 0, 2, 0, 0, 7, 0, 8],

  [2, 0, 3, 0, 0, 0, 11, 0, 10, 0, 0, 16, 14, 0, 0, 0],

  [4, 0, 11, 0, 12, 14, 0, 0, 13, 9, 0, 0, 0, 0, 0, 0],

  [0, 0, 0, 12, 0, 0, 0, 16, 0, 0, 5, 14, 0, 0, 0, 0],

  [14, 0, 16, 13, 0, 2, 1, 0, 11, 0, 0, 3, 10, 0, 0, 0],

  [3, 0, 0, 0, 0, 0, 0, 10, 15, 0, 0, 0, 0, 0, 16, 0],

  [16, 0, 0, 9, 11, 0, 0, 0, 14, 0, 6, 4, 0, 8, 13, 10],

  [10, 14, 6, 0, 16, 0, 0, 0, 0, 0, 0, 0, 15, 0, 0, 12],

  [0, 0, 0, 0, 14, 9, 0, 12, 16, 10, 0, 0, 3, 0, 0, 7],

];

Above Sudoku will be solved successfully as it meets all necessary configuration requirements of a Sudoku.

Test Case 1

var randomSudoku = [

  [11, 2, 0, 14, 3, 0, 0, 6, 7, 8, 0, 10, 0, 0, 15, 16],

  [0, 4, 0, 0, 2, 0, 0, 7, 0, 7, 16, 12, 0, 5, 14, 0],

  [0, 0, 0, 0, 10, 0, 0, 0, 1, 2, 0, 5, 0, 3, 7, 0],

  [0, 9, 7, 3, 8, 12, 0, 13, 6, 0, 14, 15, 1, 4, 10, 2],

  [0, 3, 2, 0, 5, 16, 8, 0, 0, 6, 10, 0, 0, 15, 12, 13],

  [6, 0, 8, 7, 1, 3, 0, 0, 12, 0, 15, 0, 2, 10, 0, 14],

  [13, 0, 12, 0, 9, 10, 0, 2, 4, 0, 3, 0, 16, 1, 5, 6],

  [9, 10, 14, 16, 0, 6, 12, 11, 5, 0, 2, 0, 0, 7, 0, 8],

  [2, 0, 3, 0, 0, 0, 11, 0, 10, 0, 0, 16, 14, 0, 0, 0],

  [4, 0, 11, 0, 12, 14, 0, 0, 13, 9, 0, 0, 0, 0, 0, 0],

  [0, 0, 0, 12, 0, 0, 0, 16, 0, 0, 5, 14, 0, 0, 0, 0],

  [14, 0, 16, 13, 0, 2, 1, 0, 11, 0, 0, 3, 10, 0, 0, 0],

  [3, 0, 0, 0, 0, 0, 0, 10, 15, 0, 0, 0, 0, 0, 16, 0],

  [16, 0, 0, 9, 11, 0, 0, 0, 14, 0, 6, 4, 0, 8, 13, 10],

  [10, 14, 6, 0, 16, 0, 0, 0, 0, 0, 0, 0, 15, 0, 0, 12],

  [0, 0, 0, 0, 14, 9, 0, 12, 16, 10, 0, 0, 3, 0, 0, 7],

];

Above Sudoku cannot be solved as Row 2 has two same colours, that is 7, which violates the configuration for a Sudoku.

Test Case 2

var randomSudoku = [

    [0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

    [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],

];

Another unsolvable Sudoku is as there is no information on the colours each vertex has.