```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Threading;
namespace SudokuSolutionValidator
{
    class Program
    {
        //27 Theards kiem tra Sudoku
        static Thread[] arrThread = new Thread[27];
        static bool valid = false;
        class paramaters
        {
            public int row { get; set;}
            public int column { get; set; }
        };
        //Ma tran Sudoku
        static int[,] sudoku =new int[9,9] {
                                        {6, 2, 4, 5, 3, 9, 1, 8, 7},
                                        {5, 1, 9, 7, 2, 8, 6, 3, 4},
                                        {8, 7, 7, 6, 1, 4, 2, 9, 5},
                                        {1, 4, 3, 8, 6, 5, 7, 2, 9},
                                        {9, 5, 8, 2, 4, 7, 3, 6, 1},
                                        {7, 6, 2, 3, 9, 1, 4, 5, 8},
                                        {3, 7, 1, 9, 5, 6, 8, 4, 2},
                                        {4, 9, 6, 1, 8, 2, 5, 7, 3},
                                        {2, 8, 5, 4, 7, 3, 9, 1, 6}
                                    };

        static void Main(string[] args)
        {
            DateTime startdate = DateTime.Now;
            int i, j,z, index=0;

            //9 Thread kiem tra dong
            for (i=0; i<9; i++)
            {
                arrThread[index] = new Thread(isRowValid);
                arrThread[index++].Start( new paramaters() { row = i });
                //isRowValid(new paramaters() { row = i });
            }
            //9 Thread kiem tra cot
            for (j = 0; j < 9; j++)
            {
                arrThread[index] = new Thread(isColumnValid);
                arrThread[index++].Start( new paramaters() { column = j });
                //isColumnValid(new paramaters() { column = j });
            }
            //9 Thread kiem tra 9 ma tran
            for (z = 0; z < 9; z++)
            {
                int r = (z / 3) * 3, c = (z % 3) * 3;
                arrThread[index] = new Thread(is3x3Valid);
                arrThread[index++].Start(new paramaters() { row = r, column = c });
                //is3x3Valid(new paramaters() { row = r, column = c });
            }
            //Doi 27 ket thuc
            for (index = 0; index < 27; index++)
            {
                arrThread[index].Join();
            }
            if (valid)
                Console.WriteLine("Sudoku Valid!!");
            else
                Console.WriteLine("Sudoku Invalid!!");
            DateTime finishDate = DateTime.Now;
            TimeSpan time = finishDate - startdate;
            Console.WriteLine("Total:=" + time.Ticks);
            Console.ReadLine();
        }
        static void isColumnValid(object param)
        {
            paramaters p = (paramaters)param;
            for (int i=0; i<9;i++)
            {
                for(int j=i+1; j<9;j++ )
                    if((1 > sudoku[i, p.column] || sudoku[i, p.column] > 9 ) ||(sudoku[i,p.column]== sudoku[j, p.column]))
                    {
                        Console.WriteLine("Invalid in column: " + p.column);
                        valid = false;
                        return;
                    }

            }
        }

        //Phuong thuc kiem tra dong hop le
        static void isRowValid(object param)
        {
            paramaters p = (paramaters)param;
            for (int i = 0; i < 9; i++)
            {
                for (int j = i + 1; j < 9; j++)
                    if ((1 > sudoku[p.row, i] || sudoku[p.row, i] > 9 )||
                    (sudoku[p.row, i] == sudoku[p.row, j]))
                    {
                        Console.WriteLine("Invalid in row: " + p.row);
                        valid = false;
                        return;
                    }
                }
            }
            //Phuong thuc kiem tra ma tran con hop le
            static void is3x3Valid(object param)
            {
                paramaters p = (paramaters)param;
                int[] validityArray = new int[9] { 0, 0, 0, 0, 0, 0, 0, 0, 0 }; //
                for (int i = p.row; i < p.row+3; i++)
                {
                    for(int j = p.column; j< p.column+3; j++)
                    {
                        int num = sudoku[i, j];
                        if (num < 1 || num > 9 || validityArray[num - 1] == 1)
                        {
                            Console.WriteLine("Invalid in matix("+ p.row / 3+","+
                    p.column / 3+")");
                            valid = false;
                            return;
                        }
                        validityArray[num - 1] = 1;
                    }
                }
            }
        }
    }
```