Assignment

- Title. Constraint satisfaction problem

- Problem statement:
  Implement cryptaritthematic problem or n queens problem or graph coloring problem

- Objectives:
  We will be able to
  - apply branch & bound & backtracking to solve n-queens problem

- S/W & H/W requirements:
  - OS Ubuntu / Fedora 20
  - python libraries

- Theory:

Branch and Bound

- Branch & bound is an algorithmic design paradigm for discrete and combinational optimazation problems

- It consists of a systematic enumeration of candidate solution by means of state research space the set of candidate solutions is thought of as forming a rooted tree. The algorithm explores branches of this tree

The algorithm explores branches of this tree. The algorithm which represent subsets of solutions set

Before enumerating the candidate solutions of a branch, the branch is checked against upper and lower tool estimation bounds on the optimal solution is discarded if it cannot produce a better solution that the best one found so far by the algorithm.

Backtracking:

It can be defined as an algorithm technique for solving problems recursively by trying to build a solution incrementaly one piece at a time, removing those solutions that fails to satisfy the constraints of the problems at any point of time.

N-queens problem definition:

The N-queens is a problem of placing N-queens on an N×N chessboard so that no 2 queens attack each other.

Algorithm:

```
1.) start from the leftmost corner
2.) If all queens are placed
            return true
3.) Try all rows in current column
      For every row ; do
        a.) If queen can be added placed
            safely in row then mark this
            [row, column] as post of solution
            and recursively
        b.) If placing the queen in [row, col]
            leads to solution
                  return true

        c) If placing the queen does not
            lead to sol$^n$ backtrack and
            go to step (a)
 4.) If rows have been tried,
     nothing works
            return false
```

- Testcases:

Input size	Number of sol^n
1	1
2	0
8	92
10	724

- Conclusion:

We have successfully implemente backtracking sol^n for n-queen problem.