

# Algorithm Design and Analysis

## Homework #6

Due: 11:59am January 15, 2013

### Homework submission instructions

- This homework set only contains one problem, which is a programming assignment.
- Submit your programming assignment (problem 1) to the Judgegirl System (<http://katrina.csie.ntu.edu.tw/judgegirl/>).
- In this homework set, you don't need to submit the report of the programming assignment. The score given by judgegirl would be your final grade of this homework.

## Problem 1 – Queen

### Description

There is an  $N \times N$  chessboard and  $N$  queens. Some queens have already been placed on the chessboard and you cannot change their positions. In this problem, you need to find out how many ways are there for you to place other queens on the chessboard so that no two queens can attack each other. A queen can attack others in positions that are in the same row, the same column, or the same diagonal line.

For example:

(The positions that can be attacked by the queen 'Q' are marked as 'x')

```
. . . x . x . x
. . . . x x x .
x x x x x Q x x
. . . . x x x .
. . . x . x . x
. . x . . x . .
. x . . . x . .
x . . . . x . .
```

A solution for  $N=8$ :

```
. . . Q . . . .
. . . . . . Q .
. . Q . . . . .
. . . . . . . Q
. Q . . . . . .
. . . . Q . . .
Q . . . . . . .
. . . . . Q . .
```

### Input

The first line contains two integers,  $N$  and  $P$ , indicating the size of the chessboard and the number of the queens that have already been placed on the chessboard ( $4 \leq N \leq 19$ ). Each of the following  $P$  lines contains two integers,  $R_i$  and  $C_i$ , which means that there is a queen placed at the junction of the  $(R_i + 1)$ -th row

and the  $(C_i+1)$ -th column. No two queens given in the input can attack each other.

### Output

Output the number of ways to place other  $(N-P)$  queens in a single line. The answer will not be too large. The TA's program can solve every test case within 5 seconds in JudgeGirl.

### Sample Input

```
7 1
6 3
```

### Sample Output

```
6
```

### Hint

1. You must solve the problem with your program (not by hands). For example, if you only have instructions to print out the solutions in your program, you will get zero point for this problem (for both judge result).
2. You are encouraged to use OpenMP directives to implement a multithread algorithm to solve this problem. (compile with '-fopenmp' option in gcc)
3. There are 4 cores in the judge machine.
4. '#pragma omp parallel for' is enough to get full points. (Some useful clause: private, reduction)
5. Here we provides a lecture note written by the TA last year for OpenMP:  
<http://katrina.csie.ntu.edu.tw/openmp/#/>

### Grade

Judge result (100%)

