PROJECT#1

[INTRO & RECURSION & BACKTRACKING]

Due Date: January 21, 1999 (800 points)

CS 4/56101 DESIGN & ANALYSIS OF ALGORITHM

Spring 2000, Department of Math and Computer Science, Kent State University

About Projects: The objective of the projects in this class is not to test your programming language skill, rather is to give you an first hand appreciation of the performance difference that design choices in algorithms can make.

- It means that even after quite an excellent implementation, you may end up with very poor grade if you do not write a report according to all the performance analysis requested. Grader will not be able to judge your work, no matter how good it is, without the analysis.

- It also means, you can use any language in which you are fluent. However, for helping the grading, all codes have to be submitted in the form that it is at least 'demonstrable'.

- A select few of you will be invited to demonstrate your program to the class either by running them from a PC or logging on to a unix terminal from my laptop on the Project Day (May 1st). Therefore it is encouraged that you make your program verbose. Let it print out some explanation as it progresses and show few statistics, time, and any other data that may help viewers to appreciate its main performance characteristics. This however, does not mean that a graphical interface is required.

The best made demonstrable projects will be kept in the course web-site. If your project qualifies for that you will get extra 10% bonus on the overall project grade.

Project:

1. (a) Complete the **Eight-Queens** program on your computer and generalize it for **n-queen** so that **n** number of queens can be placed in an **nxn** board. Find out exactly how many positions are investigated by including a counter that is incremented every time the function AddQueen is started. (b) Plot a graph showing the position count against board size, for n= 4 to 16. You may reuse codes given in the course helper web-site provided you understand it completely. (200 points)

2. (a) Complete the Game of Life version 1 and 2 both. Insert time measurement functions in your codes. You may reuse codes given in the course helper web-site provided you understand it completely. You have noticed that the input output mechanism of the demonstrated version is quite awkward. Modify the code so that it helps you to load map configurations easily and runs desired number of iterations to help in your timing data collection.

   (b) By experimentation with your program verify the statement that the ***execution time for Game of Life version 1 depends on the grid size, while the execution time for the Game of Life version 2 depends on the number of cell changing their state.*** Plot appropriate graphs. (Hint: you need to compare the Execution time vs. number of change in cell states, and execution time vs. grid size for both of the algorithms.) (600 points).

How to Submit:

You can use C/C++/ Java (or any other language after consultation with me and TA). Assuming C++ implementation, In this assignment you have created a set of program files *.cc and one report **proj2.doc** which contains all your explanations and graphs. You should submit a hardcopy of your report to me on the class. You should submit a **Makefile** so that TA can easily compile them in **aegis**. Also you should submit a **readme** file which should clearly state how to run your program.

On top of each file include your name, data and project number. Add:

```
/****************************************************
Name:
Date:
Project/Question Number:
ALG 2000 SPRING
Instructor: KHAN, KSU
*****************************************************/
```

For source files (*.cc) comment them.
You now need to mail all of these files to TA into one package using the following procedure:

1. create a zip file using command (this will create a compressed file named **project1.zip**)
   `%zip -r project1 yourprojectdirectory.`

2. send email to ALG2000S@maui.mcs.kent.edu with the files **project1.zip** and **report.doc** by the deadline.

3. TA will send you a confirmation email stating the receipt of your project only when he was able to download run and install it. If you have not received the confirmation within a week contact TA immediately.

Check thoroughly before you submit. If you need to re-send, for any reason inform TA beforehand. Keep a copy of all the files including shar.project1 in your directory. Do not modify them afterward. If need arises, TA may want to check these files. Any modification afterward (reflected in the file date) will result in late submission penalty.

Cheating and Copy:

Projects have to be done individually. If a copy (from your class mate or from web) is caught, all involved submissions (original as well as the copies) will be penalized. So it is your responsibility to guard your work. Secure the read/write access of your directories. Any copy will result in ZERO grade for the assignment for both party. Only exception is when you report the theft of your work in advance.