

PROJECT#1

[INTRO & RECURSION & BACKTRACKING]

Due Date: October 5th 11:55p.m. (100 points)

CS 5929 DESIGN & ANALYSIS OF ALGORITHM

Department of Computer Science, Texas 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.
- You can implement your project using C, C++, Java under Linux or ask the grader if he would grade the project in the language that you are more familiar with.
- 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. 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.

Grader: Venkata Nagarjuna Muppa nm1167@txstate.edu

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 (how many times AddQueen function was called) against board size, for $n = 4$ to 16. You may reuse the codes from the book provided that you understand it completely. (30 points)
2. (a) Complete the Game of Life version 1 and 2 both. Insert time measurement (system call) functions in your code. 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. (30 points)

(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. **Explain those graphs**) (40 points).

How to Submit:

You can use C/C++/ Java (or any other language if the grader is OK with that). 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 **Makefile** together with your project files so that grader can easily compile them. Also submit a **readme** file which should clearly state how to run your program.

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

```
/******  
Name/team members names:  
Date:  
Project/Question Number:  
ALG CS5329  
Instructor: Komogortsev, TSU  
***** /
```

- Comment your source files
- Use Assignment link on the TRACS course web-page to upload your file.
- Be sure that your file is uploaded.
- You need to upload these files on TRACS in one package.
- You can have maximum of two people on each team, but each of you **MUST** submit the project on TRACS.
- Use following procedure to create this package.

1. create a zip with the name **project1_Name(s).zip** file using “zip” command. Include your source and documentation files and your report file in the archive. The report files inside of the archive should be named **report_YourName.doc**

2. Upload archive **project1_Name(s).zip** by the deadline on TRACS.

3. You will receive a confirmation e-mail from TRACS when your files are uploaded. If you don't receive this e-mail, double check that you have submitted everything properly! Once you have submitted your code you cannot re-submit it – please check everything carefully before submission.

Late submissions will have a penalty of 5% of the maximum project score per day.

Cheating and Copy:

Coding can be done in teams, but the report should be created for each person 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.