

AI – Project 1

Rules:

- Plagiarism of any kind will result in failure for the course. No exceptions. If you are not sure whether or not you are plagiarizing from the web, classmates, or any other person/source, it is your responsibility to clarify it with the professor before submitting the code.
- This is an individual assignment. You are not allowed to work with any other student.
- You are allowed to use the web to learn more about simulated annealing, MC, queens problem. You are not allowed to use or copy source code from the web.
- You are allowed to write the code in any programming language. You should submit all your source files and a README.txt file which should describe how to compile and run the code.
You will receive 0 points if the code does not compile, i.e., generates errors during the compilation.
- Submit a printout of your code in class. Submit the source code and the README file as one zipped directory via Blackboard.

1 Solving the Queens Problem with Hill Climbing

Given an $N \times M$ board, W white queens, and B black queens, the objective is to place all the queens on the board so that no white queen attacks a black queen.

Recall that a queen attacks another queen if it is in the same row, column, or diagonal. Note that in our setting, a white queen is allowed to be in the same row, column, or diagonal with another white queen. Similarly, a black queen is allowed to be in the same row, column, or diagonal with another black queen.

The program should be named `queens` and it should be run from the command line as follows:

```
queens N M W B tmax solution.txt
```

where

- N is the number of rows of the board,
- M is the number of columns of the board,
- W is the number of white queens,
- B is the number of black queens,
- $tmax$ is an upper bound on the runtime (in seconds), and
- `solution.txt` is the name of the output file.

The program should use hill climbing to solve the problem. Specifically, it should implement the following strategy:

```

s <-- random placement of white and black queens
while s is not a solution and time < tmax
  s_neigh <-- neighbor with the best score
  if score(s) > score(s_neigh)
    break
  else
    s <-- s_neigh
print s

```

Note that it is up to you to define the score for each placement of queens. You should define the score in such a way as to increase the performance of your program so that it can quickly find a solution to the modified queens problem.

At the end, the program should write the solution to the file `solution.txt`. The solution should show the placement of the queens on the board. For example, the board below

```

|W|W| | | |
| | | |B|B|
|W|W| | | |
| | | | | |
YES

```

shows a solution with 4 white queens and 2 black queens on a 4×5 board.

The program should print at the end of the solution file, “YES” if it has found a solution, or “NO” if it has not found a solution.

2 Requirements for Graduate-Course Enrollment

If you are taking AI as a graduate course, you also need to implement Simulated Annealing. See

https://en.wikipedia.org/wiki/Simulated_annealing

Your program should take an additional argument, `method`, which should be HC for Hill Climbing or SA for Simulated Annealing. So, the program would be run as

```
queens N M W B tmax solution.txt method
```

Make sure to cite the sources that you are using for determining an appropriate annealing schedule (how you change the temperature).

3 Extra Credit

Extra credit (+10pts) will be given to the student that has the best program, i.e., capable of solving large problem instances faster than programs submitted by other students.

For the extra credit, you can use HC or Simulated Annealing (as described in this project) or you can modify HC and Simulated Annealing in any way that you like. If you do make modifications, make sure to indicate that in the README file. You can call your method for extra credit as “extra”, so that the program can be run as

queens N M W B tmax solution.txt extra

Finally, for ranking purposes, those enrolled in CSC/EE 447 will only be ranked with respect to other students in CSC/EE 447 Those enrolled in CSC/EE 547 will be ranked with respect to all students.

So, it is possible for a student in CSC/EE 447 to get +10pts of extra credit and for a student in CSC/EE 547 to get +10pts of extra credit. If the student in CSC/EE 447 has the best program overall (including over students in CSC/EE 547), then the student in CSC/EE 447 will take +20pts of extra credit.