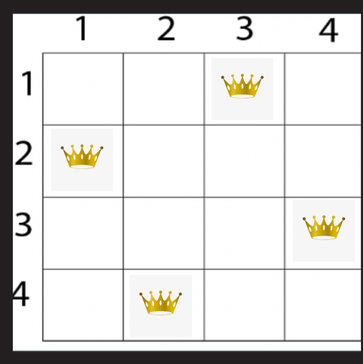
CP 468 Term Project: N-Queens

Zongyang Li, Jeetindra Dhori, Xiaocong Lian, Rini Perencsik and Khushi Satra

Group 7



N-Queens Representation

We fixed a given queen to each of the N columns.Therefore, an N-queens assignment is represented by a 1xN vector such that the ith element represents the row number of the queen in column i.

Conflicts Representation

- Row vector (1xN) such that the ith element represents the number of queens in row i
- Diagonal 1 vector (1x2N-1) such that the ith element represents the number of queens in the ith diagonal in the set of diagonals beginning at the top right corner
- Diagonal 2 vector (1x2N-1) such that the ith element represents the number of queens in the ith diagonal in the set of diagonals beginning at the top left corner

Initial State Creation

- The initial state was chosen by a greedy assignment process that assigns a queen to a row with the minimum number of conflicts. The algorithm begins by randomly placing a queen in column 1, and then for every column after that, it places a queen in the row that minimizes the number of conflicts with the current assignment. This was done so to reduce the number of steps that the min conflicts algorithm had to run.

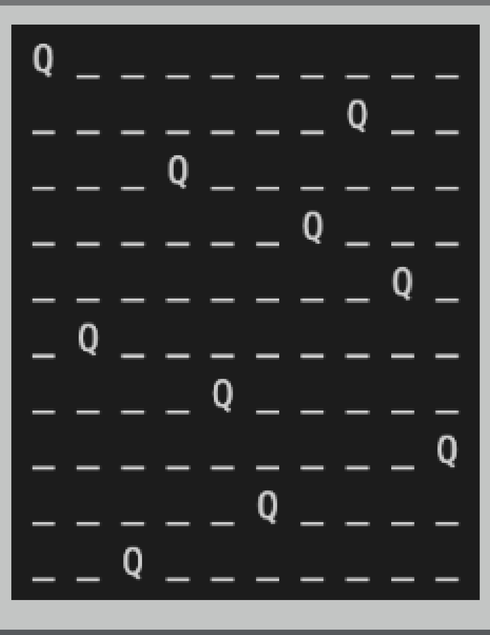
Program at high level

- Program accepts N as input
  - Initialize the initial state to an empty NxN chess board
  - Build the initial state through a greedy assignment process
    - updating puzzle and conflicts representation
  - Run min conflicts local search algorithm
    - updating puzzle and conflicts representation
- Returns
  - output.txt: vector of valid row assignments for each queen in each column
  - grid.txt: a visual representation of the N-queens solved board

Optimization challenges

- Challenge: high variance in steps and time. We wanted to guarantee that a solution could be found in a reasonable and estimable amount of time
- Reason for challenge: Min conflicts is very sensitive to initial state. But, once a good initial state has been found, it takes on average 50 steps
- Solution: create initial state through a greedy, min conflicts assignment

Solution found for N = 10



Machine Specifications

Hardware Overview:

Model Name:	MacBook Pro
Model Identifier:	MacBookPro17,1
Chip:	Apple M1
Total Number of Cores:	8 (4 performance and 4 efficiency)
Memory:	8 GB
System Firmware Version:	6723.81.1
Serial Number (system):	FVFFG9NVQ05D
Hardware UUID:	930C556E-166F-5542-9866-0E06127FD45D
Provisioning UDID:	00008103-00182920118A001E
Activation Lock Status:	Disabled

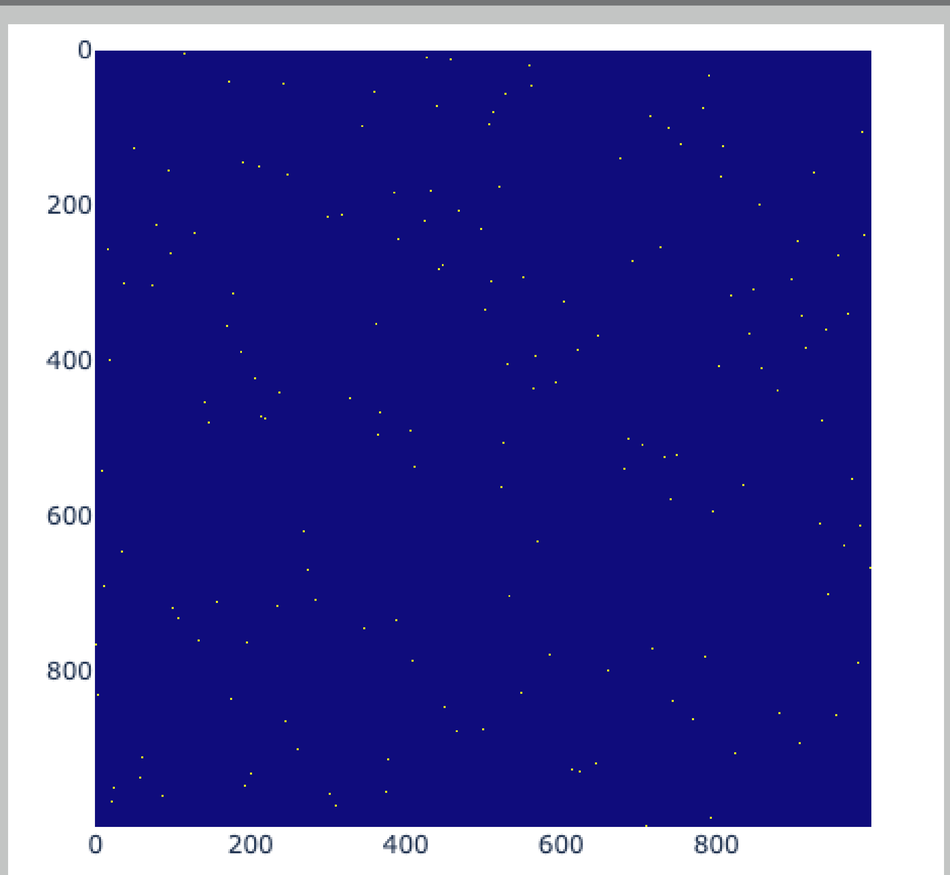
Results

N	Time to create initial state (sec)	Time to solve after initial state	Total time	Number of steps
10	0.000146	0.001758	0.001918	44
100	0.002099	0.004255	0.006371	113
10,000	3.02511	0.017272	3.0424	42
100,000	298.04	0.496	298.536	142
1,000,000	29742.5	5.0688	29747.6	105

Time Complexity Analysis

- This majority of the time it takes for our program to find a solution is spent on building a good initial state. This is because the time complexity of our initial state creation is $O(N^2)$ since for every column (xN), we iterate through every row (xN) to find the one with the minimum number of conflicts, which itself takes constant time. However, we can see that once a good initial state has been reached, the actual time it takes to find a solution is relatively quick, even for a large N (5 seconds for 1M). This is because the time it takes to find a solution is O(max_steps*N) = O(N).

Graphical representation for N = 1,000



Contact Information

- Zongyang Li (lixx7236@mylaurier.ca, 180272360)
- Jeetindra Dhori (dhor6000@mylaurier.ca, 180726000)
- Khushi Satra (satr3100@mylaurier.ca, 203383100)
- Xiaocong Lian (lian7210@mylaurier.ca, 191717210)
- Rini Perencsik (pere4930@mylaurier.ca, 191504930)