Alex Hernandez
CS 4200
Dr. Daisy Tang
12 January 2022

CS 4200 Project 2 Report

My approach:

My approach is first creating a method that counts the amount of pairs attacking each other. Then to find the best configuration branching from the current I loop through each queen and make it move in all the possible tiles it can go to. I remember the best configuration out of all of them returns that configuration in my addEdges method. For hill climbing it works from one path and does it until it solves or hits a local minimum. For min conflicts it randomly selects a new queen each time and has its own version of the addEdges method, mostly the same just working off of the one randomly chosen queen instead of looping through all of them. If min conflicts doesn't solve it in 2000 steps then we return fail.

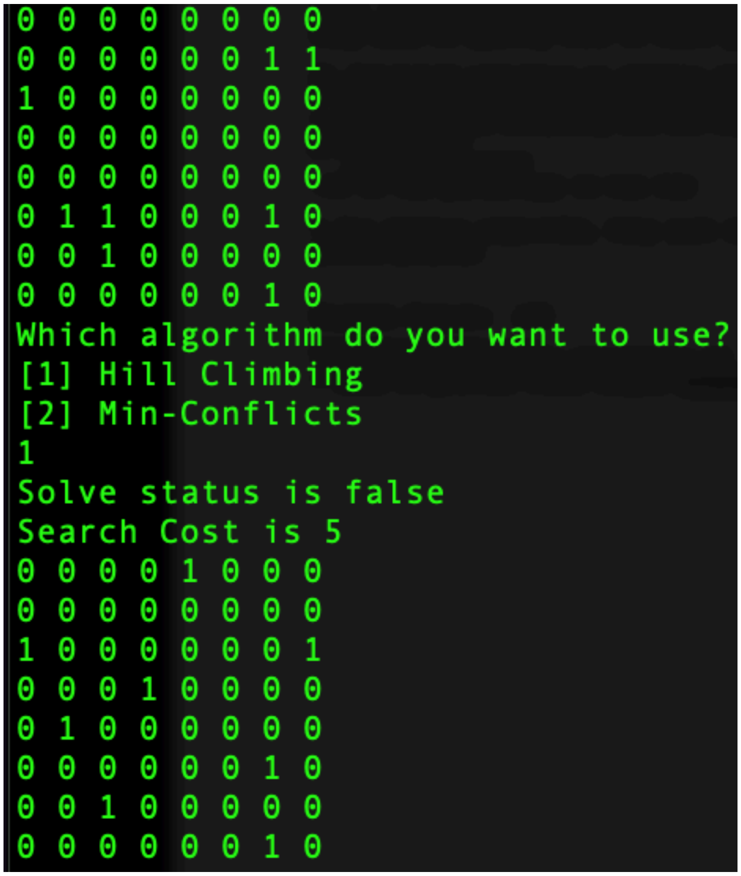
Analysis: (Done from 1000 test case method, results may slightly vary)

	Steepest Hill Climbing	**Min-Conflicts**
Average Time to Solve One Case (in milliseconds)	1	6
Average Step Cost	5	430
Average Success Percentage	16%	99%

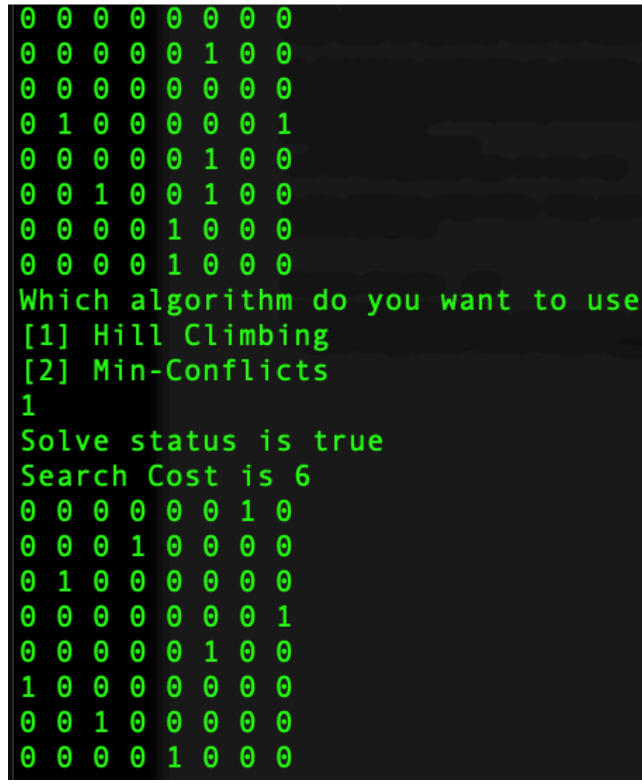
Findings:

I am not sure why I got a slightly higher success for steepest hill. In a 100 test case it ranged from 10-20 percent success, so to get an accurate result I did a 1000 test case. I based my step cost based on how many loops it took to get the results, since I didn't use any nodes for my assignment. In the 1000 case test if you run it for steepest hill it would say took average 0 milliseconds but in the 100 test case it said it did it in 1, so I used 1 for the table. In the assignment I took local search philosophy seriously and didn't care about the path to the goal just changing the current configuration. For CSP, min conflicts when I tested I saw that sometimes the algorithm took over 1000 steps but this was the far end of what it normally did. So I set max steps to 2000. Which is probably why min conflicts here as a good success percentage. It might get a hundred percent if I set max steps to 10000.

3 Sample Outputs:

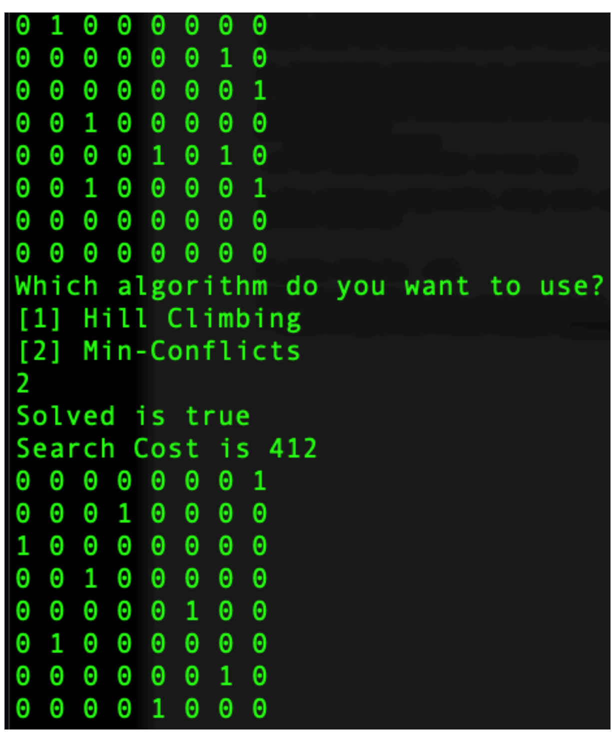


```
0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1
1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 1 1 0 0 0 1 0
0 0 1 0 0 0 0 0
0 0 0 0 0 0 1 0
Which algorithm do you want to use?
[1] Hill Climbing
[2] Min-Conflicts
1
Solve status is false
Search Cost is 5
0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 1
0 0 0 1 0 0 0 0
0 1 0 0 0 0 0 0
0 0 0 0 0 0 1 0
0 0 1 0 0 0 0 0
0 0 0 0 0 0 1 0
```



```
0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 1
0 0 0 0 0 1 0 0
0 0 1 0 0 1 0 0
0 0 0 0 1 0 0 0
0 0 0 0 1 0 0 0
Which algorithm do you want to use
[1] Hill Climbing
[2] Min-Conflicts
1
Solve status is true
Search Cost is 6
0 0 0 0 0 0 1 0
0 0 0 1 0 0 0 0
0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 1
0 0 0 0 0 1 0 0
1 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0
```

• Here we have sample outputs, 2 of hill climbing (a solved and unsolved), and a solved min conflicts.



```
0 1 0 0 0 0 0 0
0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 1
0 0 1 0 0 0 0 0
0 0 0 0 1 0 1 0
0 0 1 0 0 0 0 1
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
Which algorithm do you want to use?
[1] Hill Climbing
[2] Min-Conflicts
2
Solved is true
Search Cost is 412
0 0 0 0 0 0 0 1
0 0 0 1 0 0 0 0
1 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0
0 0 0 0 0 1 0 0
0 1 0 0 0 0 0 0
0 0 0 0 0 0 1 0
0 0 0 0 1 0 0 0
```