The Queen Game:

I spent a couple of hours on Friday hacking together a program to allow me to interactively explore dominating sets of the Queen graph.

Disclaimer: I don’t have time to put in comments or make it structured in an ideal way right now. But I am releasing it because it is fun to play with.

See the connex resources for the file Queen.java

To play with it:

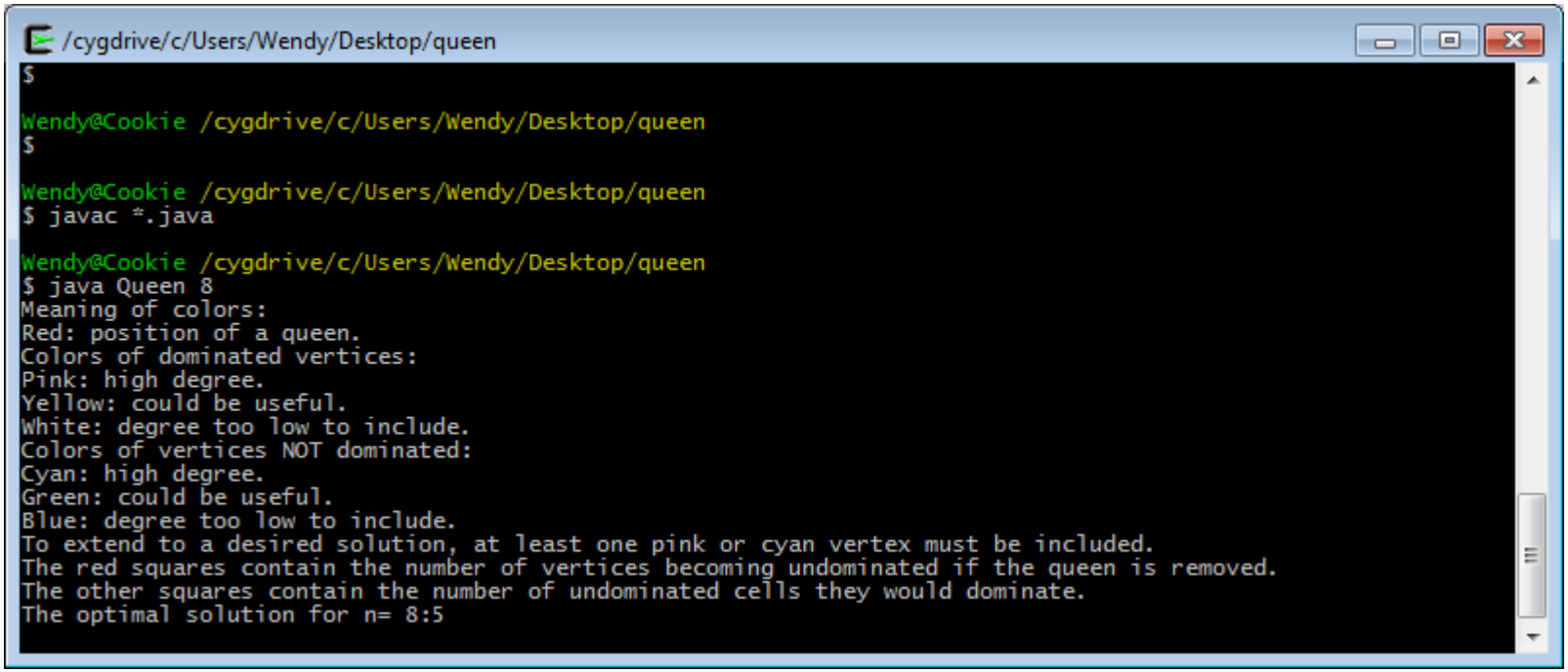
javac Queen.java

java Queen <dimension>

For example, for an 8 by 8 board:

java Queen 8

Some help text will be printed in your console window:



Red: position of a queen.

Colors of dominated vertices:

Pink: high degree.
Yellow: could be useful.
White: degree too low to include.

Colors of vertices NOT dominated:

Cyan: high degree.
Green: could be useful.
Blue: degree too low to include.

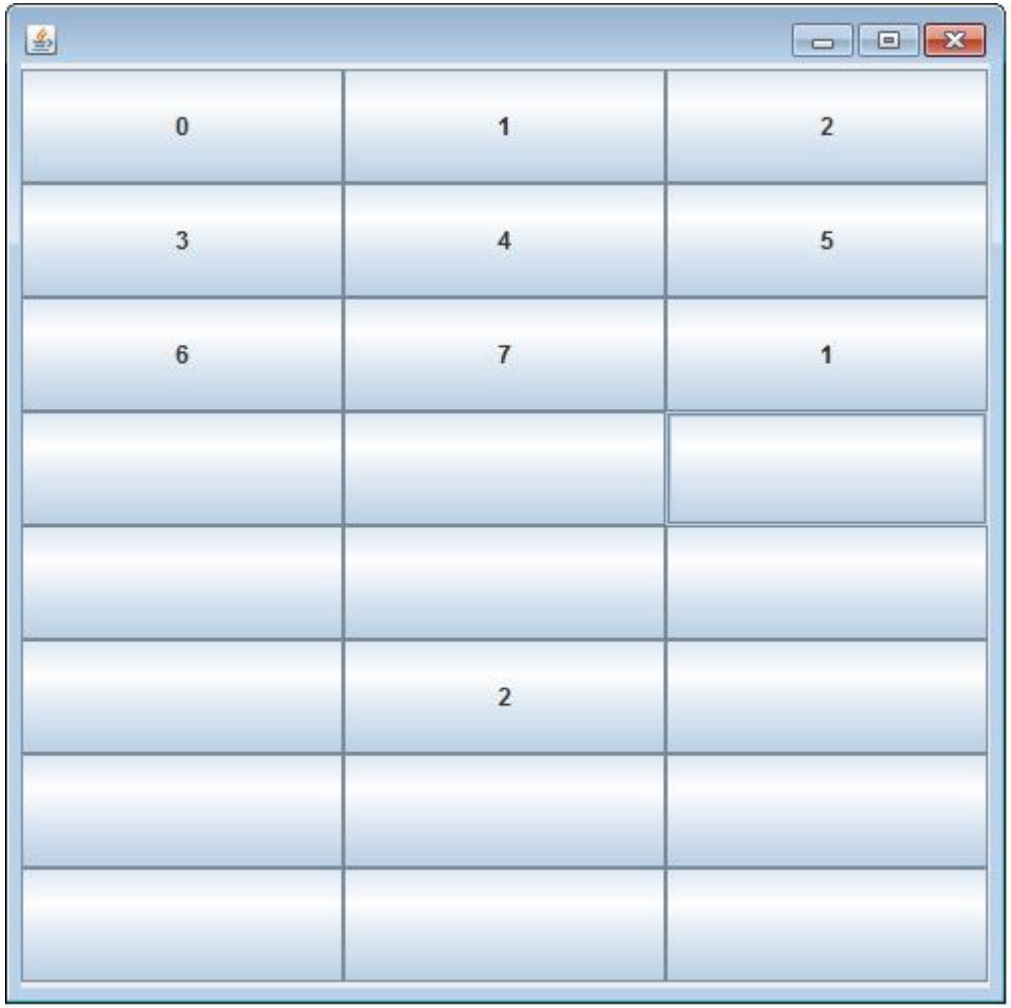
To extend to a desired solution, at least one pink or cyan vertex must be included.

The red squares contain the number of vertices becoming undominated if the queen is removed.

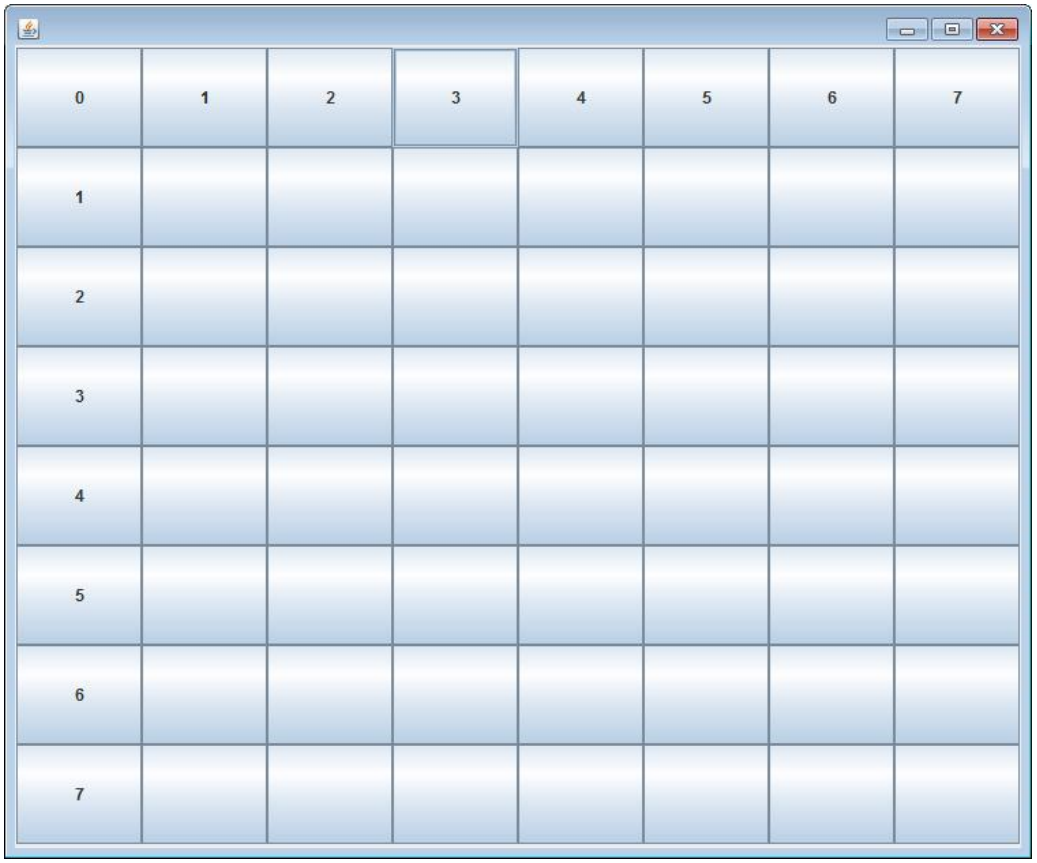
The other squares contain the number of undominated cells they would dominate.

The optimal solution for n= 8:5

The board may start out distorted:

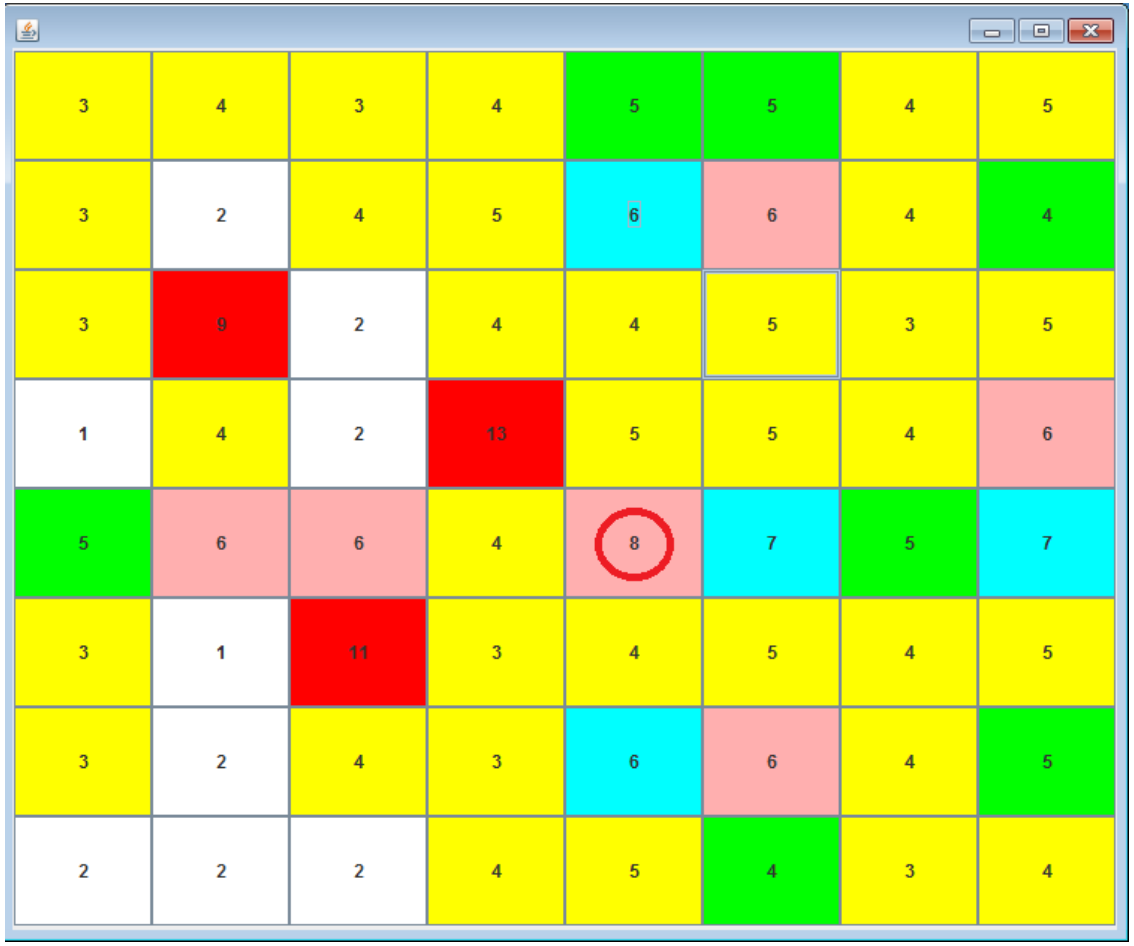


Drag the bottom righthand corner to resize it so it looks like this:

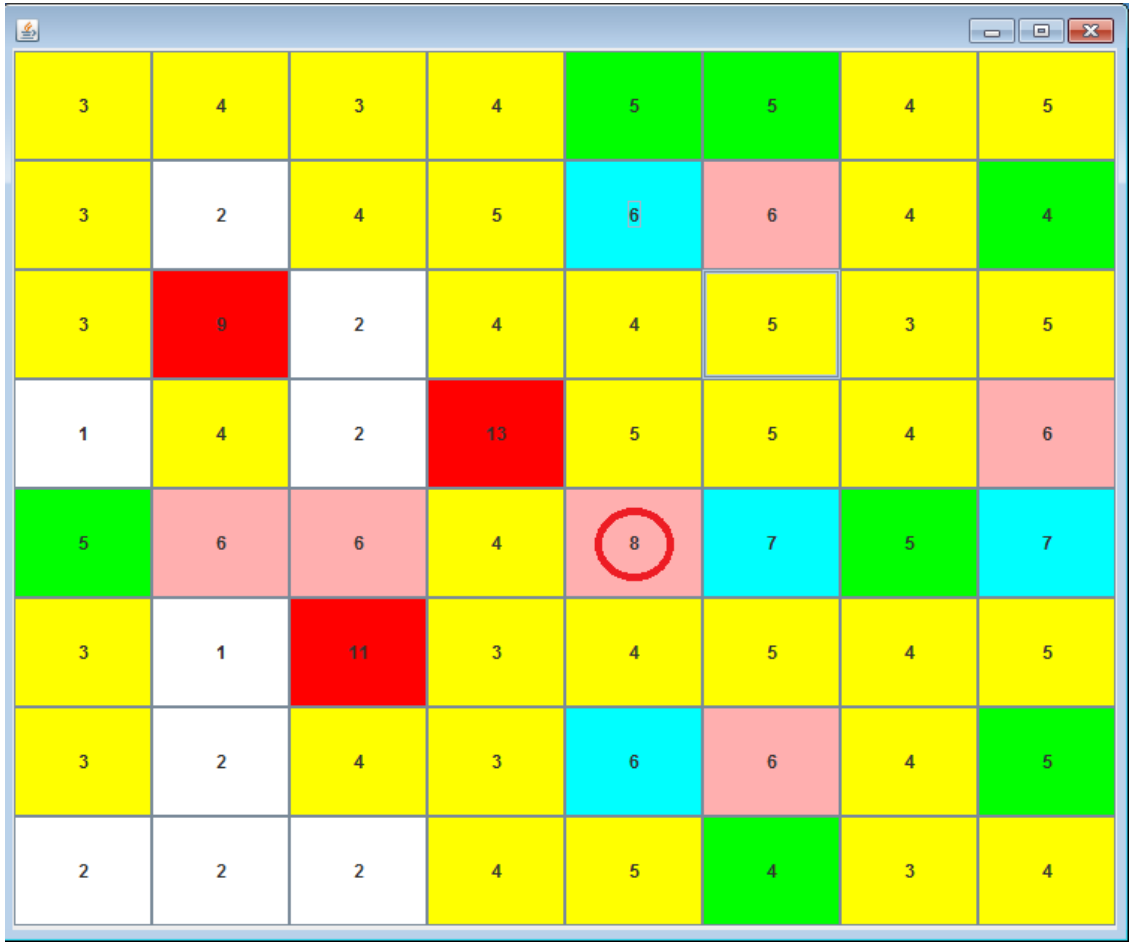


Clicking on a non-red square puts a queen there. Clicking on a queen removes it.

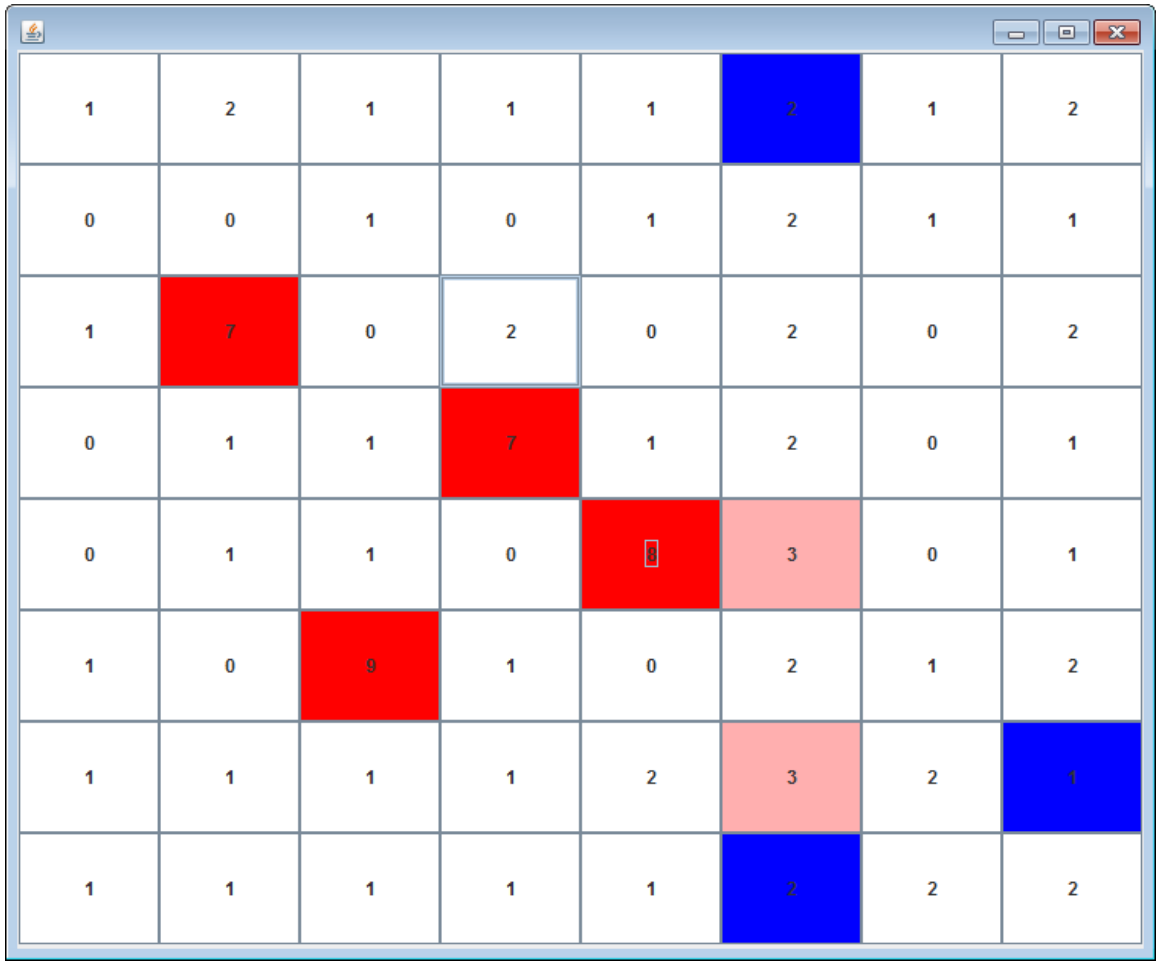
Dominated: red (queen), pink, yellow, and white.
Not dominated: blue, green, cyan.
At least one square that is pink or cyan must be included to reach an optimal solution.



If you need k more vertices in the dominating set, the console shows in reverse sorted order how many new cells might be dominated.



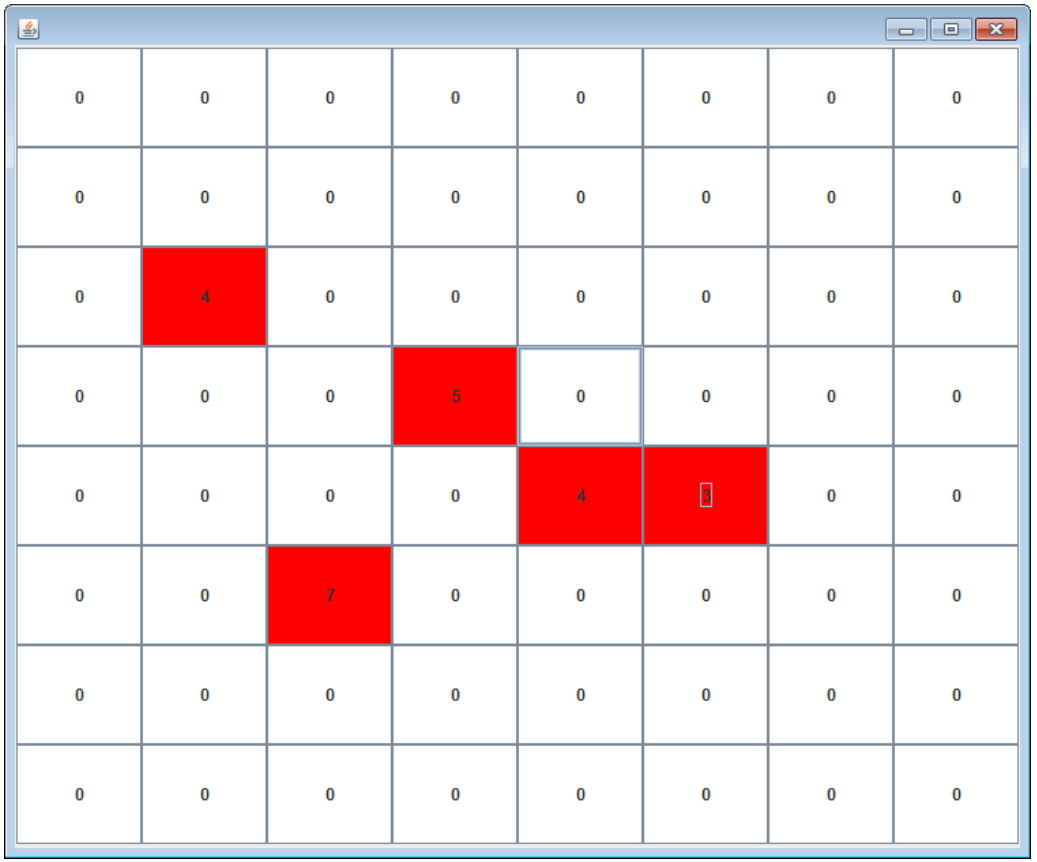

```
Sorted 61:
  8  7
Number of queens: 3
Number of cells not dominated: 11
Upper bound  : 15 using at most 2 more.
```




```
Sorted 60:
  3
Number of queens: 4
Number of cells not dominated: 3
Upper bound  : 3 using at most 1 more.
```

There is a text representation of the dominating set. If you solve an open problem, make sure you save this!

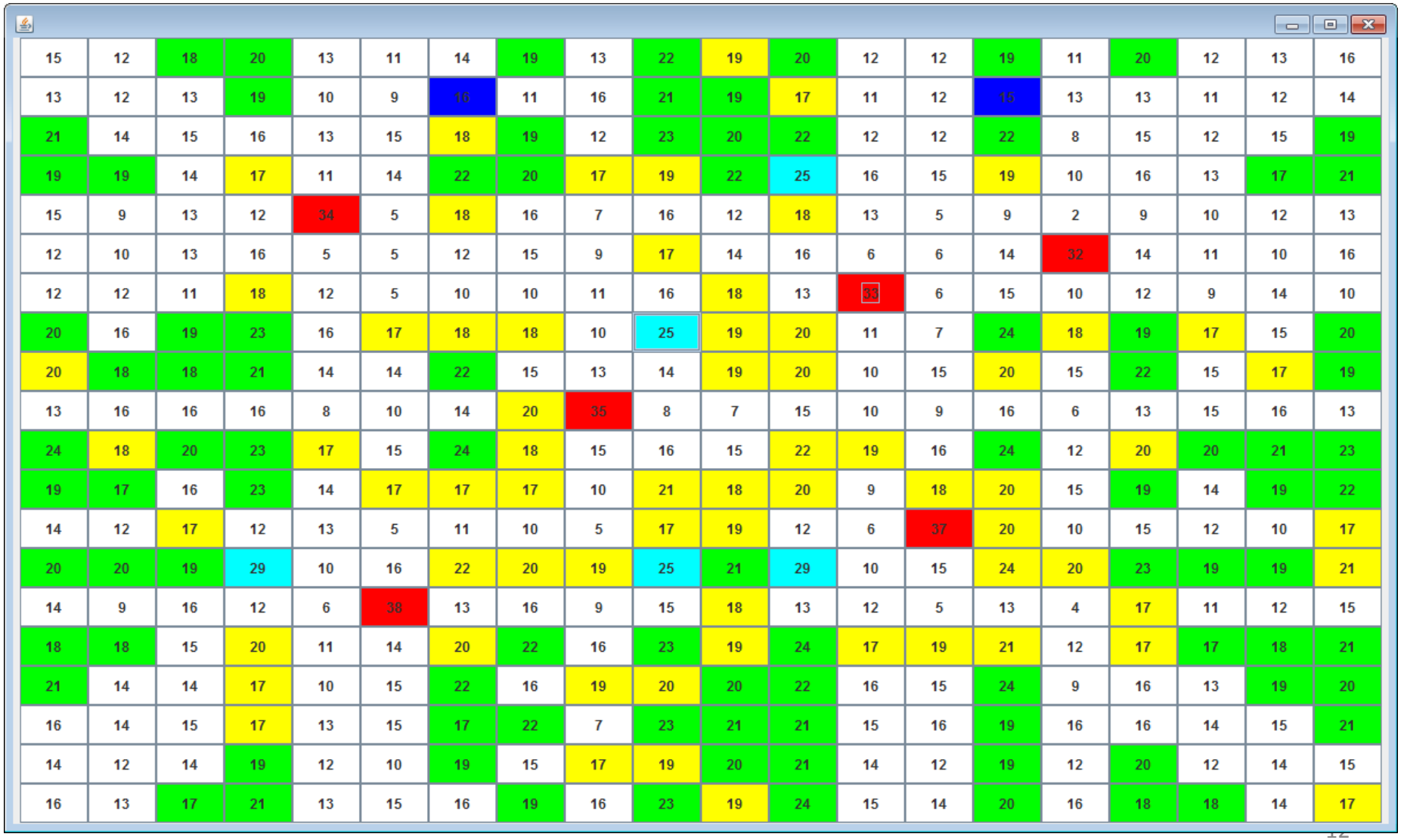
```
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
_ Q _ _ _ _ _ _
_ _ _ Q _ _ _ _
_ _ _ _ Q Q _ _
_ _ Q _ _ _ _ _
_ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _
```



Sorted 59:

Number of queens: 5
Number of cells not dominated: 0
Upper bound : 0 using at most 0 more.
GOOD DOMINATING SET!

The optimal solution for n= 20:
OPEN RESEARCH, range 10 to 11

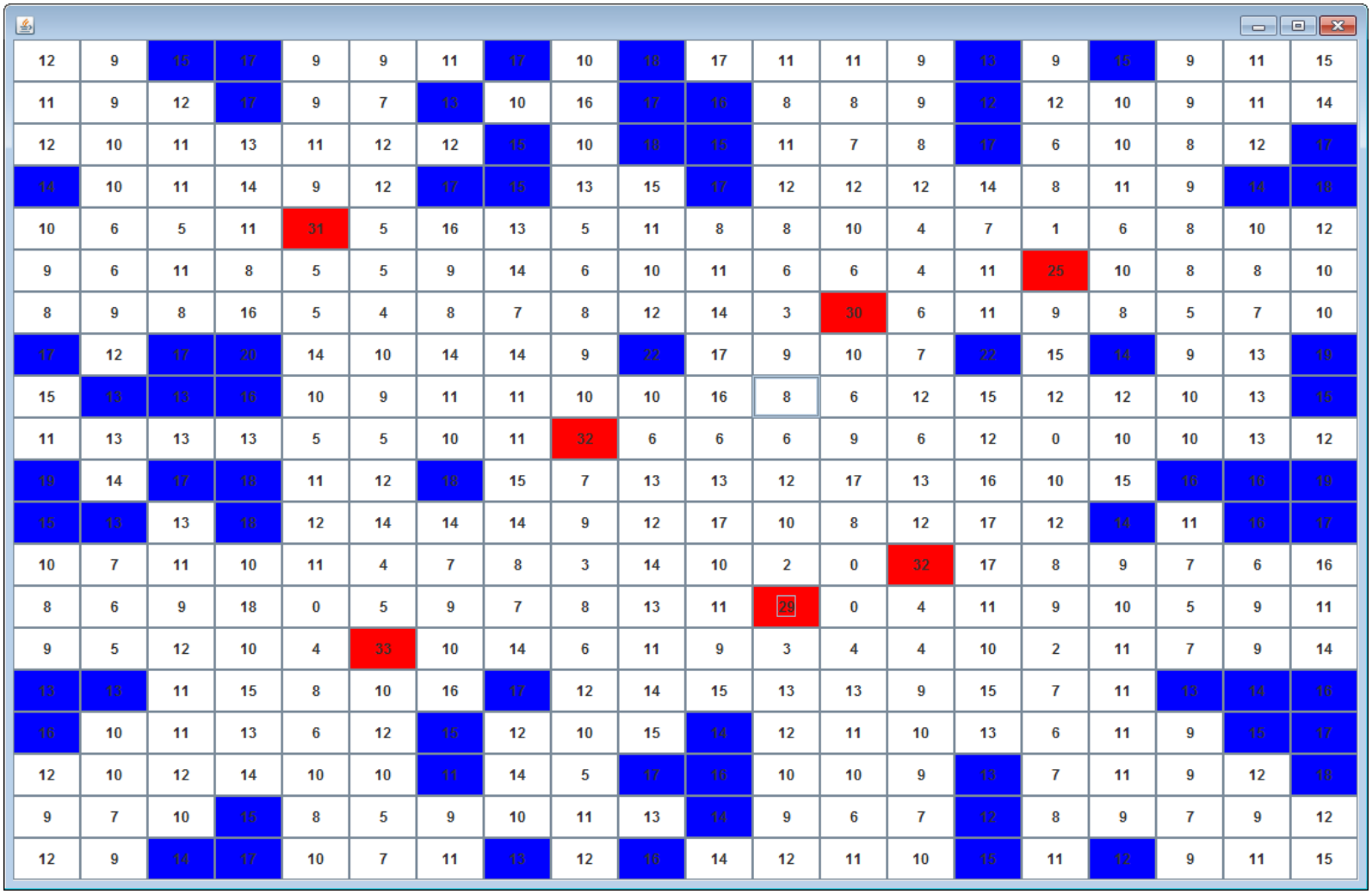


15	12	18	20	13	11	14	19	13	22	19	20	12	12	19	11	20	12	13	16
13	12	13	19	10	9	16	11	16	21	19	17	11	12	15	13	13	11	12	14
21	14	15	16	13	15	18	19	12	23	20	22	12	12	22	8	15	12	15	19
19	19	14	17	11	14	22	20	17	19	22	25	16	15	19	10	16	13	17	21
15	9	13	12	34	5	18	16	7	16	12	18	13	5	9	2	9	10	12	13
12	10	13	16	5	5	12	15	9	17	14	16	6	6	14	32	14	11	10	16
12	12	11	18	12	5	10	10	11	16	18	13	33	6	15	10	12	9	14	10
20	16	19	23	16	17	18	18	10	25	19	20	11	7	24	18	19	17	15	20
20	18	18	21	14	14	22	15	13	14	19	20	10	15	20	15	22	15	17	19
13	16	16	16	8	10	14	20	35	8	7	15	10	9	16	6	13	15	16	13
24	18	20	23	17	15	24	18	15	16	15	22	19	16	24	12	20	20	21	23
19	17	16	23	14	17	17	17	10	21	18	20	9	18	20	15	19	14	19	22
14	12	17	12	13	5	11	10	5	17	19	12	6	37	20	10	15	12	10	17
20	20	19	29	10	16	22	20	19	25	21	29	10	15	24	20	23	19	19	21
14	9	16	12	6	38	13	16	9	15	18	13	12	5	13	4	17	11	12	15
18	18	15	20	11	14	20	22	16	23	19	24	17	19	21	12	17	17	18	21
21	14	14	17	10	15	22	16	19	20	20	22	16	15	24	9	16	13	19	20
16	14	15	17	13	15	17	22	7	23	21	21	15	16	19	16	16	14	15	21
14	12	14	19	12	10	19	15	17	19	20	21	14	12	19	12	20	12	14	15
16	13	17	21	13	15	16	19	16	23	19	24	15	14	20	16	18	18	14	17

The high degree vertices seem to often line up in rows or columns or on a diagonal. If you select one of them, it greatly decreases the number of vertices another one on the same row, column or diagonal will dominate.

Note also:
There will be some cells not dominated that cannot be dominated with one of the high degree vertices. Hence our simple upper bound is not as tight as it could be.

After 7 vertices have been added, the bounds are usually strong enough to tell us we cannot complete the dominating set:



12	9	15	17	9	9	11	17	10	18	17	11	11	9	13	9	15	9	11	15
11	9	12	17	9	7	13	10	16	17	16	8	8	9	12	12	10	9	11	14
12	10	11	13	11	12	12	15	10	18	15	11	7	8	17	6	10	8	12	17
14	10	11	14	9	12	17	15	13	15	17	12	12	12	14	8	11	9	14	18
10	6	5	11	31	5	16	13	5	11	8	8	10	4	7	1	6	8	10	12
9	6	11	8	5	5	9	14	6	10	11	6	6	4	11	25	10	8	8	10
8	9	8	16	5	4	8	7	8	12	14	3	30	6	11	9	8	5	7	10
17	12	17	20	14	10	14	14	9	22	17	9	10	7	22	15	14	9	13	19
15	13	13	16	10	9	11	11	10	10	16	8	6	12	15	12	12	10	13	15
11	13	13	13	5	5	10	11	32	6	6	6	9	6	12	0	10	10	13	12
19	14	17	18	11	12	18	15	7	13	13	12	17	13	16	10	15	16	16	19
15	13	13	18	12	14	14	14	9	12	17	10	8	12	17	12	14	11	16	17
10	7	11	10	11	4	7	8	3	14	10	2	0	32	17	8	9	7	6	16
8	6	9	18	0	5	9	7	8	13	11	29	0	4	11	9	10	5	9	11
9	5	12	10	4	33	10	14	6	11	9	3	4	4	10	2	11	7	9	14
13	13	11	15	8	10	16	17	12	14	15	13	13	9	15	7	11	13	14	16
16	10	11	13	6	12	15	12	10	15	14	12	11	10	13	6	11	9	15	17
12	10	12	14	10	10	11	14	5	17	16	10	10	9	13	7	11	9	12	18
9	7	10	15	8	5	9	10	11	13	14	9	6	7	12	8	9	7	9	12
12	9	14	17	10	7	11	13	12	16	14	12	11	10	15	11	12	9	11	15

Sorted 393:
  22 22 20

Number of queens: 7

Number of cells not dominated: 71

Upper bound:
64 using at most 3 more.

CANNOT COMPLETE TO A DESIRED SOLUTION.

I did not try to make my code fast for the Queen game. Since a person is playing it and we respond relatively slowly (as compared to a computer), there was no need to include sophisticated algorithms.

Correctness was more important because I only had a couple of hours free to develop, test, debug and play with the program.

More efficient data structures might make a complete search for dimension 20 (the first open case) feasible.