MineSweeper

Member Name	NETID	RUID	Class Section
Linchen Xie	lx100	188004552	Section 1
Ningyuan Zhang	nz146	186002871	Section 1
Weikang Li	wl494	189006134	Section 2
Shenao yan	sy558	187003007	Section 1

1.Questions and Writeup

• Representation: How did you represent the board in your program, and how did you represent the information / knowledge that clue cells reveal?
Inheriting the experience from the last assignment(Maze Runner), we still use a matrix to represent the board. As you can see in the BoardGenerator class, given the value of board height, width and mine density we can get a (height*width) matrix, where 1 means a mine. We can also get a matrix counting the number of neighbors which are mines for each node, which conceals Important information. If there's a mine under a certain cell, we won't compute it and return -1 directly. We will compute the whole counting matrix at the very beginning of the program, and when we need the information of certain node, we can get the responding value directly.

• Inference: When you collect a new clue, how do you model / process / compute the information you gain from it? i.e., how do you update your current state of knowledge based on that clue? Does your program deduce everything it can from a given clue before continuing? If so, how can you be sure of this, and if not, how could you consider improving it?
Upon exploring a new node, we can easily get the number of it's nearby mines.
If the number is equal to 0, we can conclude that all neighbors all safe and add them into the self.cleared_nodes set.
If the number is greater than 0,we can store the number and all possible neighbors into the self.constraints set.

A	B	safe
C	3	safe
safe	safe	D

Figure 1: example1

Which is shown in Figure 1, there are 3 mines nearby. And except for those safe nodes, we get 4 potential nodes:A,B,C,D. We can express such condition as a equation: A+B+C+D=3, and each variable can be 0 (means it's safe) or 1(means it's a mine). We combine the node information(A,B,C,D) and the mine number(3) as a clue and store it in a set. What we need to do next is to explore another node to obtain more clues until we can reach a conclusion.
Suppose that if we can get another constraint that B+C+D=3, and compare it with the above one, then it's obvious that A=0, and B,C,D are equal to 1.
In general, what we do is convert the information of each node into an equation, and store it in a set.In this example, we store ((A,B,C,D),(3)) and ((B,C,D),(3)) in the constraints set.Afterwards, we pick any two equation and try to compare them. It's possible that we can get new information or even figure out certain cell: (A,B,C,D)-(B,C,D)=(A),3-3=0, from which we can conclude that A=0.
And I believe that my program deduce everything it can from a given clue before continuing. Because I let the program keep updating knowledge(comparing any two clues of all constraints) until there's no change any more.

• Decisions: Given a current state of the board, and a state of knowledge about the board, how does your program decide which cell to search next? Are there any risks, and how do you face them?
We give cleared,unexplored nodes priority since it's safer. If we have run out of all cleared nodes, then we pick a new node randomly.
There's certain risk when we pick random node because it's possible that we hit a mine.

• Performance: For a reasonably-sized board and a reasonable number of mines, include a play-by-play progression to completion or loss. Are there any points where your program makes a decision that you don't agree with? Are there any points where your program made a decision that surprised you? Why was your program able to make that decision?

The program works well except when it run out of cleared nodes. In this case, the program will pick a node randomly to explore, which generate a certain risk of stepping on a mine.
Here's what we can do to improve it. In order to avoid hitting a mine, we need to pick new nodes based on known clues instead of choosing a random node. We can give covered nodes whose uncovered neighbours' number is small higher priority. For example, in Figure 2, A is much less likely to be a mine than B since its uncovered neighbors return a small number.

1	A	1
	1	
3	B	3

Figure 2: example2

For a 10*10 board, the program works well when the number of mines less than 7. As I state before, picking random invites risk stepping on a mine.

• Performance: For a fixed, reasonable size of board, what is the largest number of mines that your program can still usually solve? Where does your program struggle?
Our algorithm can solve board with size (50, 50) and 50 mines in 30 seconds, which is efficient. In general, we can solve a board with 10% of its cells are mines easily. But when the number of mines increase, the board is hard to solve.

• Efficiency: What are some of the space or time constraints you run into in implementing this program? Are these problem specific constraints, or implementation specific constraints? In the case of implementation constraints, what could you improve on?
Since the size of board is not huge, time and space complexity are both acceptable.
If we compute it according to the codes, we can conclude that the time complexity is O(n^2) and the space complexity is O(n^3)

• Improvements: Consider augmenting your program's knowledge in the following way - when the user inputs the size of the board, they also input the total number of mines on the board. How can this information be modeled and included in your program, and used to inform action? How can you use this

information to effectively improve the performance of your program, particularly in terms of the number of mines it can effectively solve.

The number of mines can have influence on our sweeping strategy. For example, if the number of mines is really small, like 1 or 2, picking random node is safe because we're almost unlikely to come into a mine. On contrary, if there exists many mines on the board, we need to be cautious about the next node. In this case, it makes sense to spend more time to find a safest node to explore.

We can also end our algorithm earlier if we have find all mines.

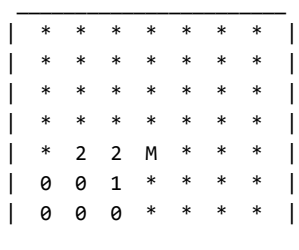
Another things we can do is when combining constraints, we just combine constraints that a related. For example, the constraints obtained for top left corner and bottom right are highly possible that they are irrelevant, so we can ignore the combination of these two, which can save us a lot of time.

Example:

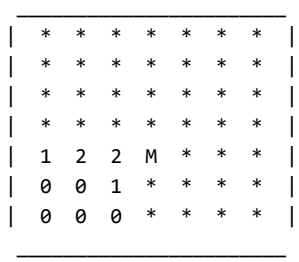
```
This is the board that I need to solve
 _____________________
|  0  0  0  0  0  0  0  |
|  1  1  2  1  1  0  0  |
|  2  M  3  M  1  0  0  |
|  M  3  M  3  3  1  1  |
|  1  2  2  M  2  M  1  |
|  0  0  1  1  2  1  1  |
|  0  0  0  0  0  0  0  |
 _____________________


 Start playing...
The node I choose is:  (6, 0)
 _____________________
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  0  *  *  *  *  *  *  |
 _____________________

The node I choose is:  (5, 1)
 _____________________
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  0  *  *  *  *  *  |
|  0  *  *  *  *  *  *  |
 _____________________
```

```
The node I choose is:  (6, 1)
 _____________________
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  0  *  *  *  *  *  |
|  0  0  *  *  *  *  *  |
 _____________________

The node I choose is:  (5, 0)
 _____________________
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  0  0  *  *  *  *  *  |
|  0  0  *  *  *  *  *  |
 _____________________

The node I choose is:  (5, 2)
 _____________________
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  0  0  1  *  *  *  *  |
|  0  0  *  *  *  *  *  |
 _____________________

The node I choose is:  (6, 2)
 _____________________
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  0  0  1  *  *  *  *  |
|  0  0  0  *  *  *  *  |
 _____________________

The node I choose is:  (4, 2)
 _____________________
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  2  *  *  *  *  |
|  0  0  1  *  *  *  *  |
|  0  0  0  *  *  *  *  |
 _____________________
```

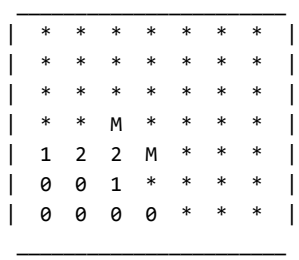
The node I choose is: (4, 1)



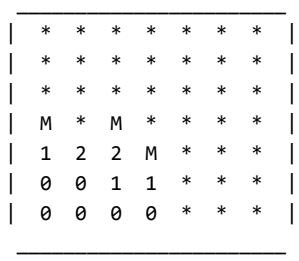
The node I choose is: (4, 0)



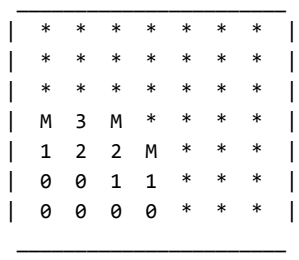
The node I choose is: (6, 3)



The node I choose is: (5, 3)



The node I choose is: (3, 1)



```
The node I choose is:  (3, 3)
 _____________________
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  M  3  M  3  *  *  *  |
|  1  2  2  M  *  *  *  |
|  0  0  1  1  *  *  *  |
|  0  0  0  0  *  *  *  |
 _____________________

The node I choose is:  (6, 4)
 _____________________
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  M  3  M  3  *  *  *  |
|  1  2  2  M  *  *  *  |
|  0  0  1  1  *  *  *  |
|  0  0  0  0  0  *  *  |
 _____________________

The node I choose is:  (5, 4)
 _____________________
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  M  3  M  3  *  *  *  |
|  1  2  2  M  *  *  *  |
|  0  0  1  1  2  *  *  |
|  0  0  0  0  0  *  *  |
 _____________________

The node I choose is:  (4, 4)
 _____________________
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  M  3  M  3  *  *  *  |
|  1  2  2  M  2  M  *  |
|  0  0  1  1  2  *  *  |
|  0  0  0  0  0  *  *  |
 _____________________

The node I choose is:  (5, 5)
 _____________________
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  M  3  M  3  *  *  *  |
|  1  2  2  M  2  M  *  |
|  0  0  1  1  2  1  *  |
|  0  0  0  0  0  *  *  |
 _____________________
```

```
The node I choose is:  (6, 5)
 _____________________
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  M  3  M  3  *  *  *  |
|  1  2  2  M  2  M  *  |
|  0  0  1  1  2  1  *  |
|  0  0  0  0  0  0  *  |
 _____________________

The node I choose is:  (3, 4)
 _____________________
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  M  3  M  3  3  *  *  |
|  1  2  2  M  2  M  *  |
|  0  0  1  1  2  1  *  |
|  0  0  0  0  0  0  *  |
 _____________________

The node I choose is:  (3, 5)
 _____________________
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  M  3  M  3  3  1  *  |
|  1  2  2  M  2  M  *  |
|  0  0  1  1  2  1  *  |
|  0  0  0  0  0  0  *  |
 _____________________

The node I choose is:  (5, 6)
 _____________________
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  M  3  M  3  3  1  *  |
|  1  2  2  M  2  M  *  |
|  0  0  1  1  2  1  1  |
|  0  0  0  0  0  0  *  |
 _____________________

The node I choose is:  (4, 6)
 _____________________
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  M  *  *  *  |
|  M  3  M  3  3  1  *  |
|  1  2  2  M  2  M  1  |
|  0  0  1  1  2  1  1  |
|  0  0  0  0  0  0  *  |
 _____________________
```

```
The node I choose is:  (6, 6)
 _____________________
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  M  *  *  *  |
|  M  3  M  3  3  1  *  |
|  1  2  2  M  2  M  1  |
|  0  0  1  1  2  1  1  |
|  0  0  0  0  0  0  0  |
 _____________________

The node I choose is:  (2, 5)
 _____________________
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  M  *  0  *  |
|  M  3  M  3  3  1  *  |
|  1  2  2  M  2  M  1  |
|  0  0  1  1  2  1  1  |
|  0  0  0  0  0  0  0  |
 _____________________

The node I choose is:  (2, 6)
 _____________________
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  M  *  0  0  |
|  M  3  M  3  3  1  *  |
|  1  2  2  M  2  M  1  |
|  0  0  1  1  2  1  1  |
|  0  0  0  0  0  0  0  |
 _____________________

The node I choose is:  (2, 4)
 _____________________
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  M  1  0  0  |
|  M  3  M  3  3  1  *  |
|  1  2  2  M  2  M  1  |
|  0  0  1  1  2  1  1  |
|  0  0  0  0  0  0  0  |
 _____________________

The node I choose is:  (3, 6)
 _____________________
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  *  M  1  0  0  |
|  M  3  M  3  3  1  1  |
|  1  2  2  M  2  M  1  |
|  0  0  1  1  2  1  1  |
|  0  0  0  0  0  0  0  |
 _____________________
```

```
The node I choose is:  (2, 2)
 _____________________
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  *  *  |
|  *  *  3  M  1  0  0  |
|  M  3  M  3  3  1  1  |
|  1  2  2  M  2  M  1  |
|  0  0  1  1  2  1  1  |
|  0  0  0  0  0  0  0  |
 _____________________

The node I choose is:  (1, 5)
 _____________________
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  0  *  |
|  *  *  3  M  1  0  0  |
|  M  3  M  3  3  1  1  |
|  1  2  2  M  2  M  1  |
|  0  0  1  1  2  1  1  |
|  0  0  0  0  0  0  0  |
 _____________________

The node I choose is:  (1, 6)
 _____________________
|  *  *  *  *  *  *  *  |
|  *  *  *  *  *  0  0  |
|  *  *  3  M  1  0  0  |
|  M  3  M  3  3  1  1  |
|  1  2  2  M  2  M  1  |
|  0  0  1  1  2  1  1  |
|  0  0  0  0  0  0  0  |
 _____________________

The node I choose is:  (1, 4)
 _____________________
|  *  *  *  *  *  *  *  |
|  *  *  *  *  1  0  0  |
|  *  *  3  M  1  0  0  |
|  M  3  M  3  3  1  1  |
|  1  2  2  M  2  M  1  |
|  0  0  1  1  2  1  1  |
|  0  0  0  0  0  0  0  |
 _____________________

The node I choose is:  (1, 3)
 _____________________
|  *  *  *  *  *  *  *  |
|  *  *  *  1  1  0  0  |
|  *  *  3  M  1  0  0  |
|  M  3  M  3  3  1  1  |
|  1  2  2  M  2  M  1  |
|  0  0  1  1  2  1  1  |
|  0  0  0  0  0  0  0  |
 _____________________
```

```
The node I choose is:  (0, 6)
 ______________________
|  *  *  *  *  *  *  0  |
|  *  *  *  1  1  0  0  |
|  *  *  3  M  1  0  0  |
|  M  3  M  3  3  1  1  |
|  1  2  2  M  2  M  1  |
|  0  0  1  1  2  1  1  |
|  0  0  0  0  0  0  0  |
 ______________________

The node I choose is:  (0, 5)
 ______________________
|  *  *  *  *  *  0  0  |
|  *  *  *  1  1  0  0  |
|  *  *  3  M  1  0  0  |
|  M  3  M  3  3  1  1  |
|  1  2  2  M  2  M  1  |
|  0  0  1  1  2  1  1  |
|  0  0  0  0  0  0  0  |
 ______________________

The node I choose is:  (0, 4)
 ______________________
|  *  *  *  *  0  0  0  |
|  *  *  *  1  1  0  0  |
|  *  *  3  M  1  0  0  |
|  M  3  M  3  3  1  1  |
|  1  2  2  M  2  M  1  |
|  0  0  1  1  2  1  1  |
|  0  0  0  0  0  0  0  |
 ______________________

The node I choose is:  (0, 3)
 ______________________
|  *  *  *  0  0  0  0  |
|  *  *  *  1  1  0  0  |
|  *  *  3  M  1  0  0  |
|  M  3  M  3  3  1  1  |
|  1  2  2  M  2  M  1  |
|  0  0  1  1  2  1  1  |
|  0  0  0  0  0  0  0  |
 ______________________

The node I choose is:  (1, 2)
 ______________________
|  *  *  *  0  0  0  0  |
|  *  *  2  1  1  0  0  |
|  *  *  3  M  1  0  0  |
|  M  3  M  3  3  1  1  |
|  1  2  2  M  2  M  1  |
|  0  0  1  1  2  1  1  |
|  0  0  0  0  0  0  0  |
 ______________________
```

```
The node I choose is:  (0, 2)
 _______________________
|  *  *  0  0  0  0  0  |
|  *  *  2  1  1  0  0  |
|  *  *  3  M  1  0  0  |
|  M  3  M  3  3  1  1  |
|  1  2  2  M  2  M  1  |
|  0  0  1  1  2  1  1  |
|  0  0  0  0  0  0  0  |
 _______________________

The node I choose is:  (0, 1)
 _______________________
|  *  0  0  0  0  0  0  |
|  *  *  2  1  1  0  0  |
|  *  *  3  M  1  0  0  |
|  M  3  M  3  3  1  1  |
|  1  2  2  M  2  M  1  |
|  0  0  1  1  2  1  1  |
|  0  0  0  0  0  0  0  |
 _______________________


  ************  well done!  ************
 _______________________
|  0  0  0  0  0  0  0  |
|  1  1  2  1  1  0  0  |
|  2  M  3  M  1  0  0  |
|  M  3  M  3  3  1  1  |
|  1  2  2  M  2  M  1  |
|  0  0  1  1  2  1  1  |
|  0  0  0  0  0  0  0  |
 _______________________
Do you want to see the Chain of Influence ? (y or n): y


Chain of Influence
This is a randomly selected start point
(6, 0)  ->  (5, 1)
(6, 0)  ->  (6, 1)
(6, 0)  ->  (5, 0)
(5, 1)  ->  (5, 2)
(5, 1)  ->  (6, 2)
(5, 1)  ->  (4, 2)
(5, 1)  ->  (4, 1)
(5, 1)  ->  (4, 0)
(5, 2)  ->  (4, 3)
(6, 2)  ->  (6, 3)
(6, 2)  ->  (5, 3)
(6, 3)  ->  (6, 4)
(6, 3)  ->  (5, 4)
(6, 4)  ->  (5, 5)
(6, 4)  ->  (6, 5)
(5, 5)  ->  (5, 6)
(5, 5)  ->  (4, 6)
(5, 5)  ->  (6, 6)
(5, 4)  ->  (4, 5)
```

```
(5, 3)  ->  (4, 4)
(4, 4)  ->  (3, 4)
(4, 4)  ->  (3, 5)
(3, 4)  ->  (2, 0)
(3, 4)  ->  (2, 3)
(3, 4)  ->  (2, 2)
(2, 2)  ->  (0, 1)
(2, 2)  ->  (2, 0)
(0, 1)  ->  (1, 0)
(0, 1)  ->  (0, 0)
(3, 5)  ->  (2, 5)
(3, 5)  ->  (2, 6)
(3, 5)  ->  (2, 4)
(3, 5)  ->  (3, 6)
(2, 5)  ->  (1, 5)
(2, 5)  ->  (1, 6)
(2, 5)  ->  (1, 4)
(1, 5)  ->  (0, 6)
(1, 5)  ->  (0, 5)
(1, 5)  ->  (0, 4)
(1, 4)  ->  (0, 3)
(2, 4)  ->  (1, 3)
(1, 3)  ->  (1, 2)
(1, 3)  ->  (0, 2)
(1, 2)  ->  (0, 1)
(1, 2)  ->  (2, 1)
(0, 1)  ->  (1, 0)
(0, 1)  ->  (0, 0)
(0, 2)  ->  (1, 1)
(4, 2)  ->  (3, 1)
(4, 2)  ->  (3, 3)
(3, 1)  ->  (2, 0)
(3, 3)  ->  (2, 0)
(3, 3)  ->  (2, 2)
(2, 2)  ->  (0, 1)
(2, 2)  ->  (2, 0)
(0, 1)  ->  (1, 0)
(0, 1)  ->  (0, 0)
(4, 1)  ->  (3, 0)
(4, 1)  ->  (3, 2)
(4, 1)  ->  (3, 1)
(4, 1)  ->  (3, 3)
(3, 1)  ->  (2, 0)
(3, 3)  ->  (2, 0)
(3, 3)  ->  (2, 2)
(2, 2)  ->  (0, 1)
(2, 2)  ->  (2, 0)
(0, 1)  ->  (1, 0)
(0, 1)  ->  (0, 0)
(4, 0)  ->  (3, 2)
(4, 0)  ->  (3, 1)
(4, 0)  ->  (3, 3)
(3, 1)  ->  (2, 0)
(3, 3)  ->  (2, 0)
(3, 3)  ->  (2, 2)
(2, 2)  ->  (0, 1)
(2, 2)  ->  (2, 0)
```

```
(0, 1)  ->  (1, 0)
(0, 1)  ->  (0, 0)
```

2. Bonus: Chains of Influence

• Based on your model and implementation, how can you characterize and build this chain of influence? Hint: What are some 'intermediate' facts along the chain of influence?

When we deduce a cell is safe of a mine, we set its parents to those nodes from which we arrived at this deduction. When we are not able to tell a cell is safe or a mine, we will put a constraint into constraints list with a 'parents' set, and this 'parents' set contains cells who deduced this constraint.

There are 4 cases:

1. When we select a cell randomly, and this cell shows '0', which means all its neighbors are safe, we can set all its neighbors' parent to this cell.
2. When we select a cell randomly, and this cell shows '9', which means all its neighbors are mines, we can set all its neighbors' parent to this cell.
3. When we combine constraints, we also union the 'parents' set of these constraints. And when the number of mines of a certain constraint is 0, we can immediately say that the cells of this constraint are all safe. In this case, we set parents of these cells to 'parents' set of this constraint.
4. Just like case 3, When we find the number of mines of a certain constraint equals to the number of cells of the constraint, we can tell that all cells are mines. In this case, we set parents of these cells to 'parents' set of this constraint.

In this way, we can build Chains of influence easily. The chain will never broken unless we run out of information or we finish the inference.

• What influences or controls the length of the longest chain of influence when solving a certain board?

We think The length of the longest chain of influence can be influenced in 3 ways:

1. Search strategy. If we search with DFS, we can build a long chain since we will start for a cell and explore the map as deep as we can. When the map is sparse, which means it has less mines, we can build a very long chain. However, when we use BFS to search, we will start from a cell and spread in all directions, rather than go deeper in one direction, so the chain can not as long as chain of DFS.
2. The information we can get when playing. If we can continue get enough information, the chain will not broken, therefore, it can be very long. However, in some cases, we cannot get enough information to deduce. In this case, we have

to pick a cell randomly and build a new chain from this cell, therefore, we will not get a very long chain but several short chain instead.

3. Distribution of mines. If the distribution of mines are sparse, we will march almost all cells to find mines, which means that we can get a very long chain. If all mines are just in a small area, in some cases, when we find all mines in this area, algorithm ends. So we will not get a long chain.

• How does the length of the chain of influence influence the efficiency of your solver?

We think that in some aspect, the length of the chain have something to do with the efficiency. If we solve a map with a long chain, it means that we do a lot of inferences when we solve the map, and it is not necessary in some cases. But a long chain also tells us that we are not run out of information when inferring, which means that we do not need to randomly select a cell to get more information.

• Experiment. Can you find a board that yields particularly long chains of influence? How does this vary with the total number of mines?

```
 ____________________________________
|  0  0  1  M  1  0  0  0  1  M  |
|  1  1  1  1  2  1  1  0  1  1  |
|  M  1  0  0  2  M  2  1  1  1  |
|  1  1  0  0  2  M  3  2  M  1  |
|  1  1  0  0  1  1  2  M  2  1  |
|  M  2  1  0  1  1  2  1  1  0  |
|  3  M  2  0  1  M  2  1  1  0  |
|  2  M  4  2  2  1  2  M  2  1  |
|  1  2  M  M  1  1  2  3  4  M  |
|  0  1  2  2  1  1  M  2  M  M  |
 ____________________________________
```

In above Board, we can see that the mines are located in nearly every corner of the map. In this case, the chain of influence is very long. In some aspect, more mines, the longer the chain of influence is.

Although the number of mines have something to do with the length of chain of influence, it is not the only factor that influences the length of chain. For example, if a map contains too much mines, we cannot deduce information easily and have to randomly select a cell in some cases, which increase the possibility of pick up a mine.

• Experiment. Spatially, how far can the influence of a given cell travel?
Very far. In some case, may constraints are dependent, if we know just result of one constraint, we can deduce all others.

• Can you use this notion of minimizing the length of chains of influence to inform the decisions you make, to try to solve the board more efficiently?
If we use BFS, we can reduce the length of chain of influence. Because, we try to spread in all directions and try to get information in this area, Therefore, the mines can not found more efficiently.

• Is solving minesweeper hard?
It dependents. If we can get enough information when inferring, it is not hard. But if we need to select cells randomly to obtain more information, we will have higher chance to lose, in which case it is hard.

3.Bonus: Dealing with Uncertainty

When the information I receive is accurate, but it is uncertain when I will receive the information, there are two solutions. The first one is if we receive nothing about certain node, we can totally ignore it, mark it unexplored and keep exploring the next node. The second one is to keep exploring the same node until I get information. Because the information must be correct, it doesn't matter that I try it more times. As long as I try enough times, I can still get the exact information I want.
When the clue has some probability of underestimating the number of surrounding mines, Clues are always optimistic. In my opinion, I can explore every node more than 1 time, and keep the maximum returned number as the final result. For example, if the probability of getting correct number is 0.8, and I keep exploring 4 times, the probability of getting underestimated information is only 0.0016, which is small enough.
When the clue has some probability of overestimating the number of surrounding mines, clues are always cautious.I can still explore every node more than 1 time, and keep the minimum returned number as the final result. For example, if the probability of getting correct number is 0.8, and I keep exploring 4 times, the probability of getting overestimated information is only 0.0016, which is small enough.