How Complicated is Minesweeper?

Richard Kaye

http://web.mat.bham.ac.uk/R.W.Kaye/

R.W.Kaye@bham.ac.uk

Birmingham University

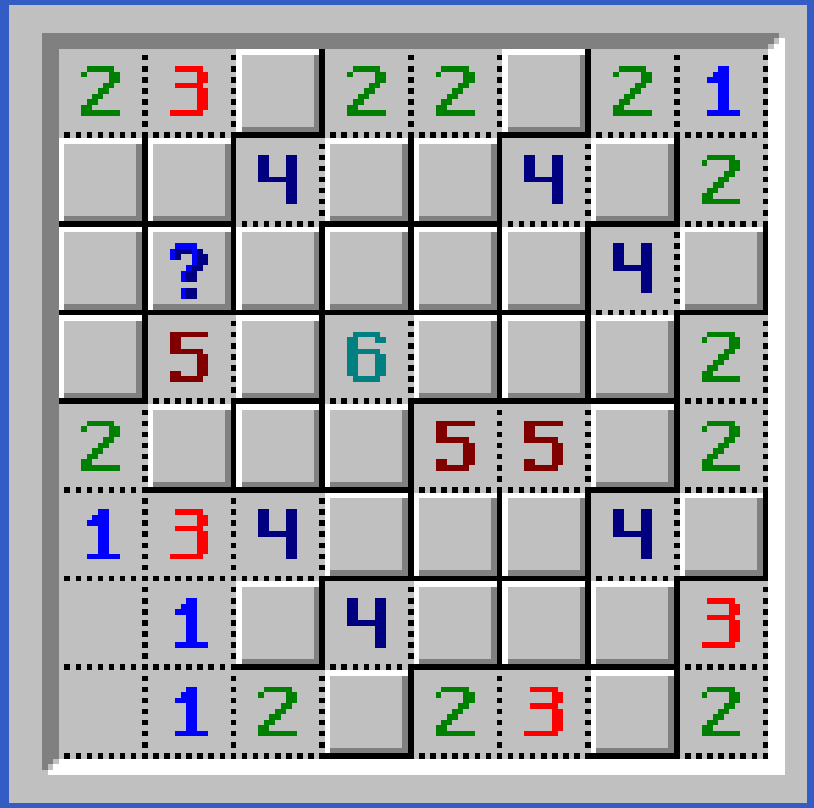
Minesweeper

- Minesweeper is a familiar computer game requiring you to locate the mines in a minefield without being blown up.
- When skilfully played, the task can be completed without having to take many risky guesses!

Minesweeper

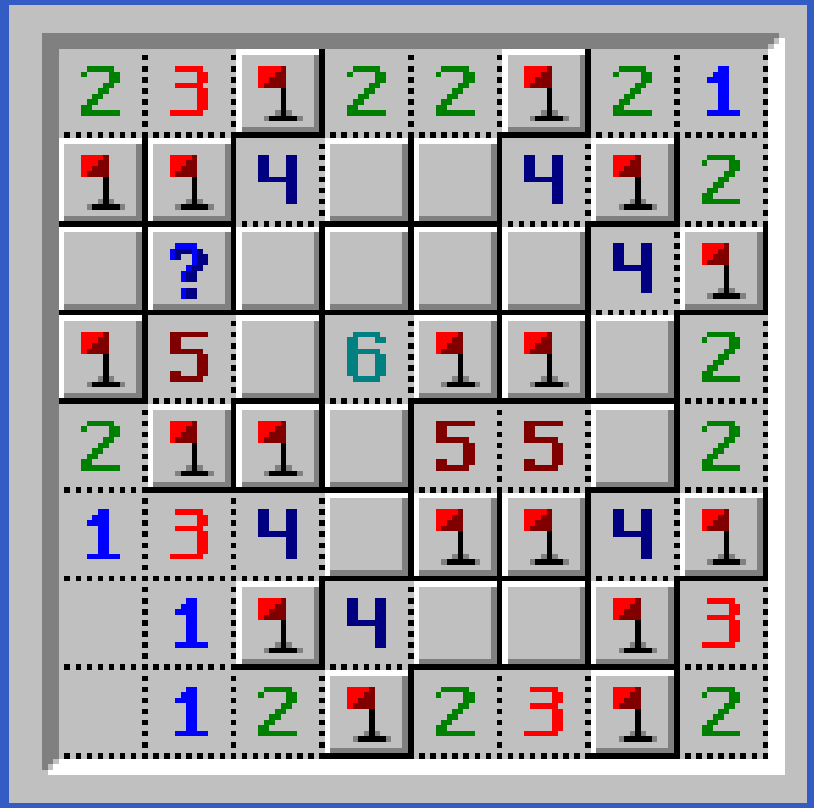
- Minesweeper is a familiar computer game requiring you to locate the mines in a minefield without being blown up.
- When skilfully played, the task can be completed without having to take many risky guesses!
- In fact the complexity of minesweeper is related to an important unsolved problem in mathematics, which in turn relates to cracking codes on the internet.

Playing the game



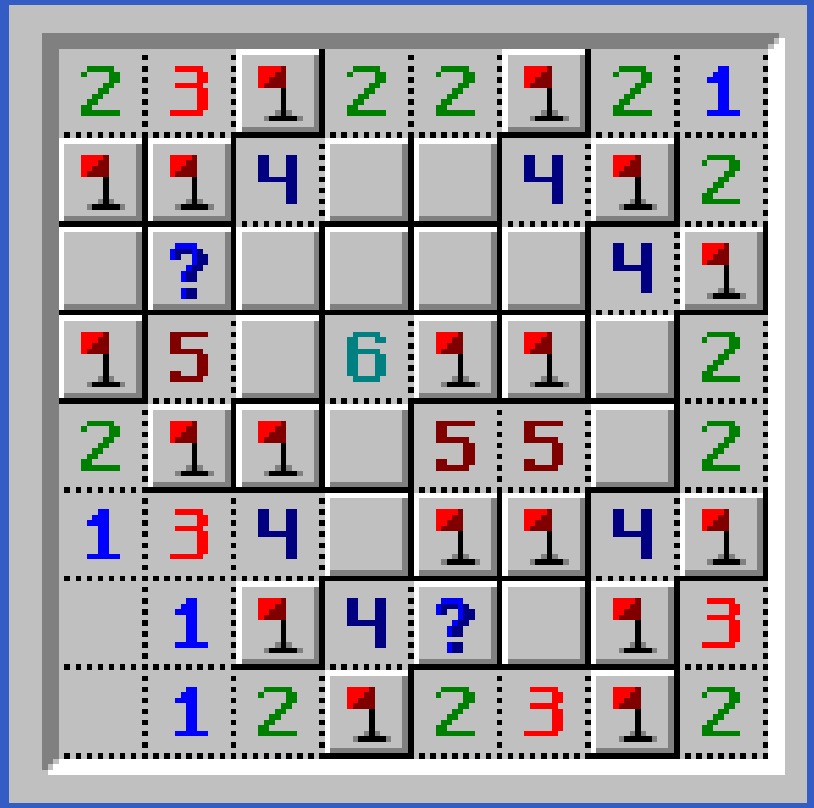
Does (2,6) have a mine?

Playing the game



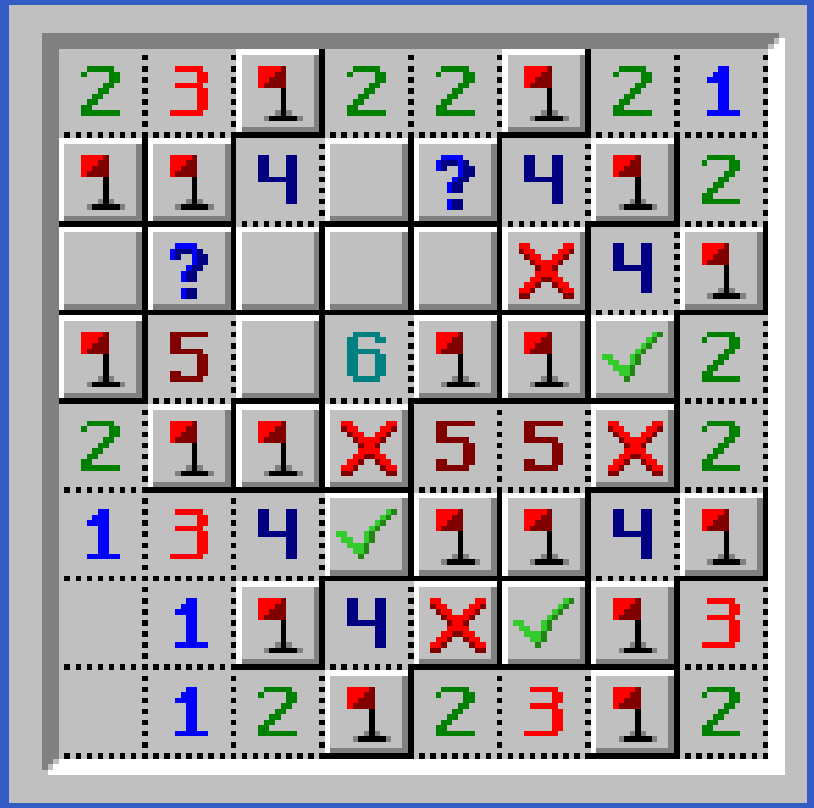
These must have mines

Playing the game



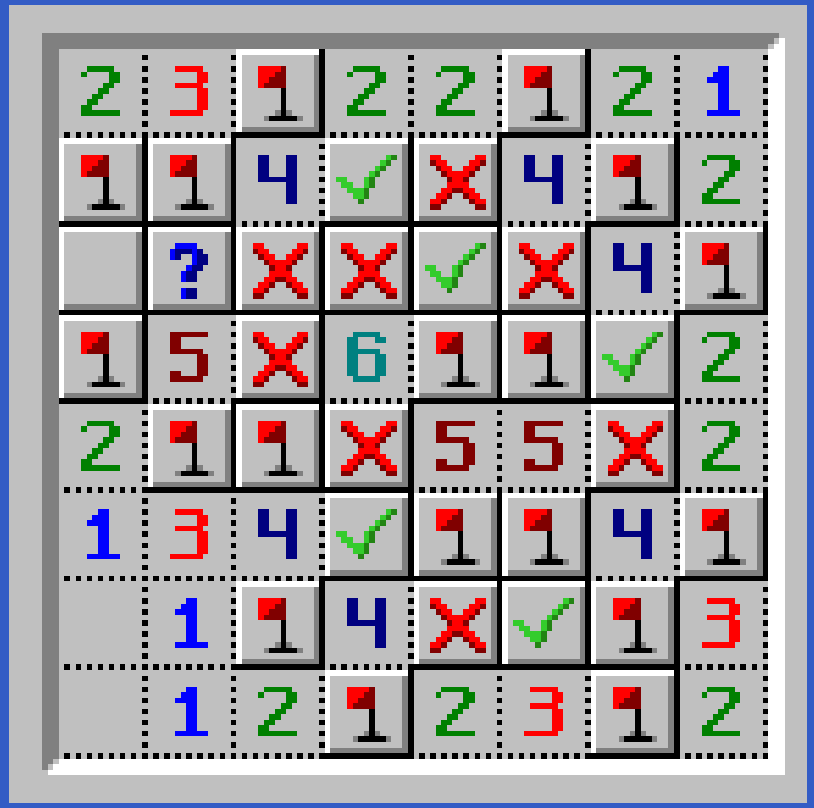
Does (5,2) have a mine?

Playing the game



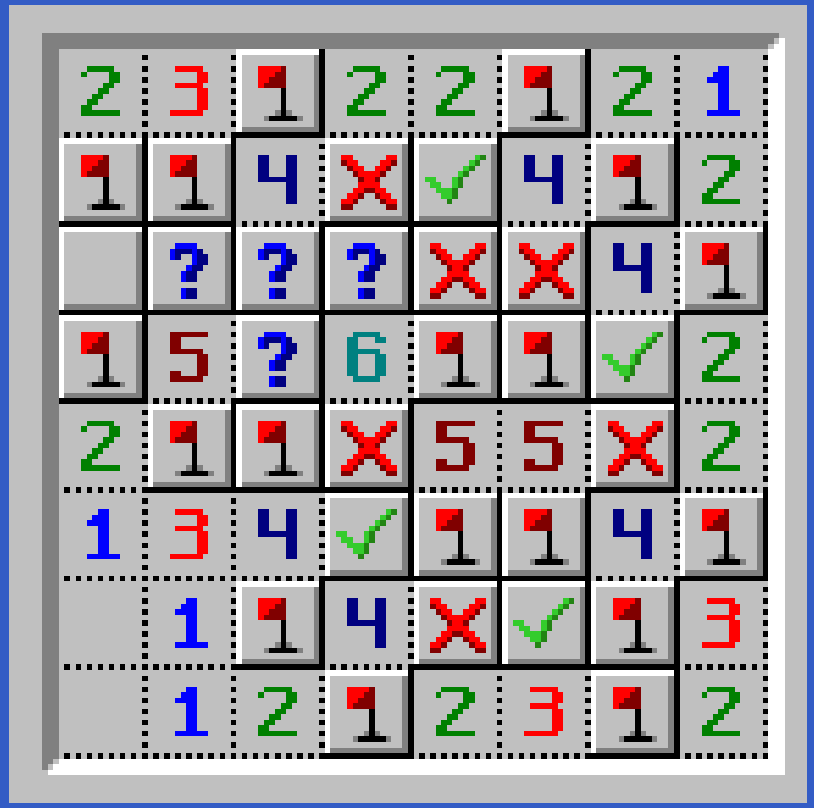
If so, does (5,7) have a mine?

Playing the game



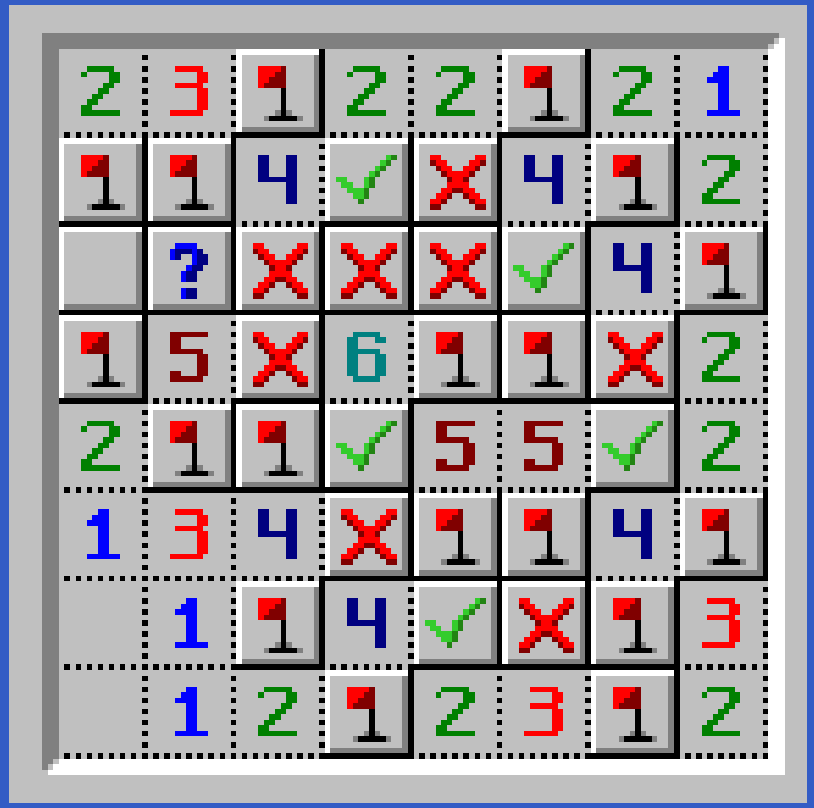
If (5,2) and (5,7) have mines

Playing the game



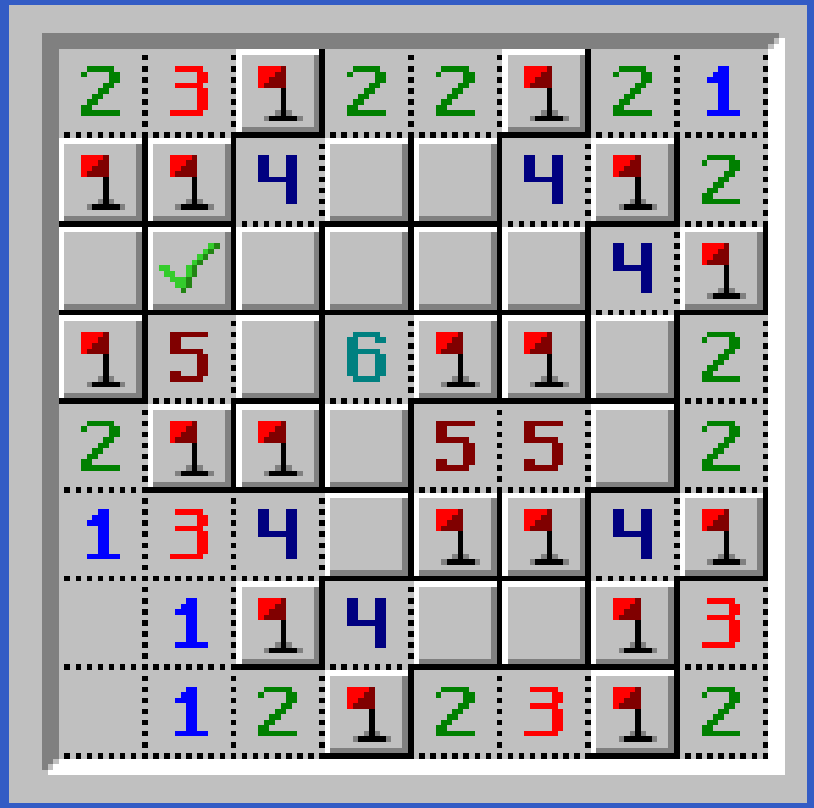
If (5,2) has a mine but (5,7) hasn't

Playing the game



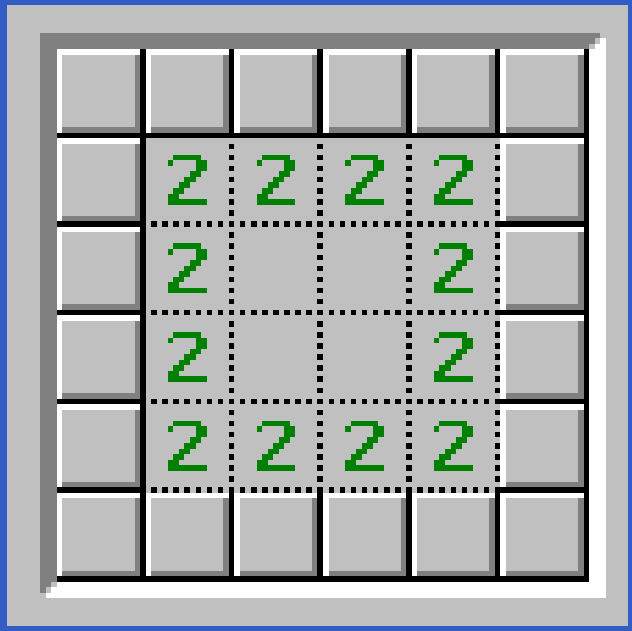
If (5,2) is clear

Playing the game



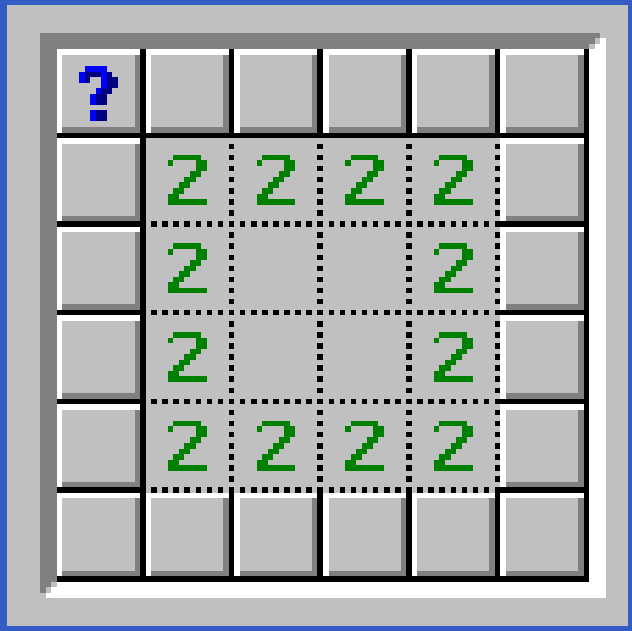
So (2,6) is clear!

A puzzle



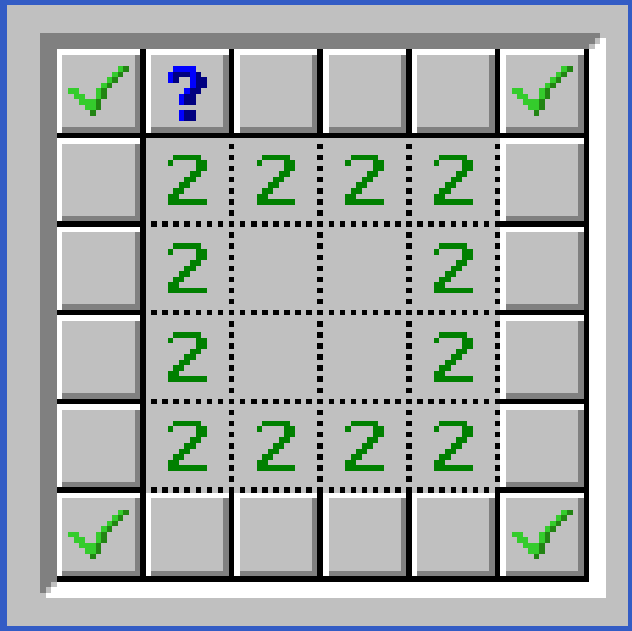
Determine the location of all mines!

A puzzle



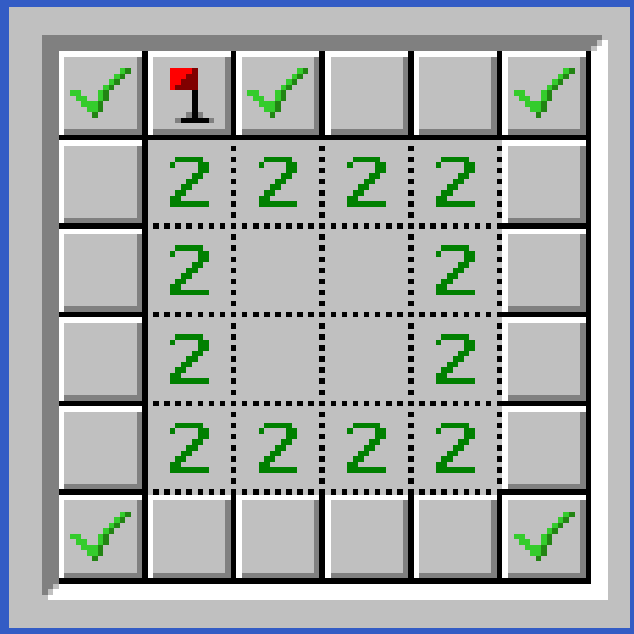
Can this be a mine?

A puzzle



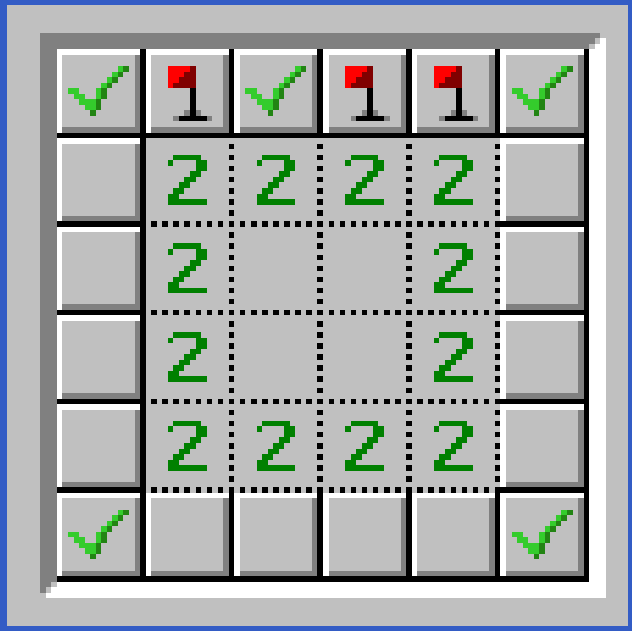
If this is a mine the one next to it is clear. . .

A puzzle



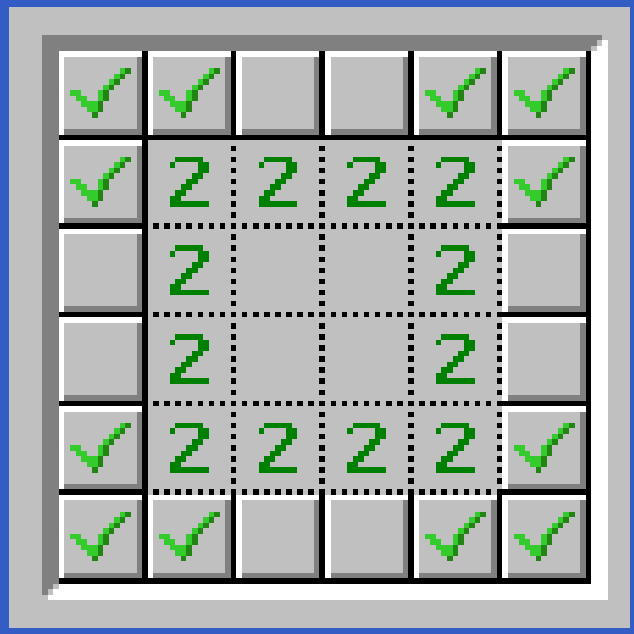
So. . .

A puzzle



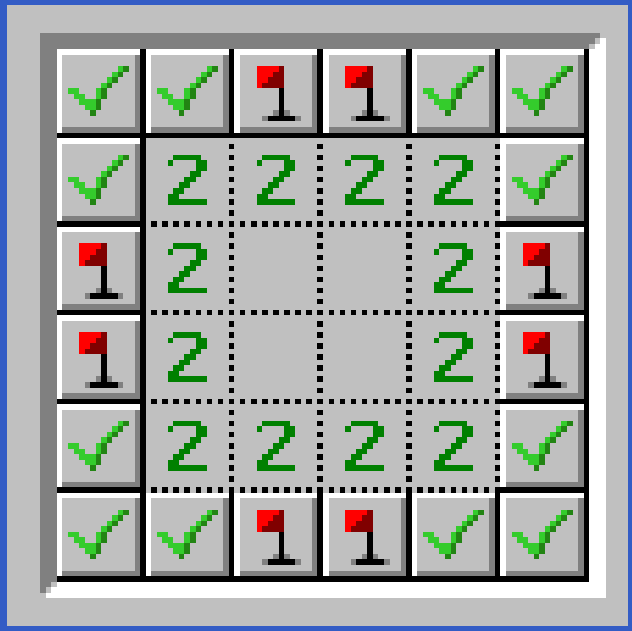
. . . which is impossible!

A puzzle



Therefore. . .

A puzzle



Solved!

What is the question?

- Minesweeper appears to be difficult to play well. How difficult is it?

What is the question?

- Minesweeper appears to be difficult to play well. How difficult is it?
- To formulate this question precisely we need to specify what we mean by playing ‘well’. We can at least assume a good player will not take stupid risks:

What is the question?

- Minesweeper appears to be difficult to play well. How difficult is it?
- To formulate this question precisely we need to specify what we mean by playing ‘well’. We can at least assume a good player will not take stupid risks:
- *We want to play in such a way that we never make a risky move when there is some square which can be uncovered safely.*

The consistency problem

- Given a minesweeper configuration, is it *consistent*? I.e., could it have arisen from some pattern of mines?

The consistency problem

- Given a minesweeper configuration, is it *consistent*? I.e., could it have arisen from some pattern of mines?
- To determine if a square is free of any mines change the configuration marking it with a mine. Then ask if the result is consistent. If not, it is safe to clear the square!

How might we solve it?

- Problems like this are quite general and require a *computer algorithm* or *computer program* as the solution.

How might we solve it?

- Problems like this are quite general and require a *computer algorithm* or *computer program* as the solution.
- There *is* an algorithm that solves this problem and solves the consistency problem. All you need to do is go though all the different combinations for the mines on the minefield in turn.

So what's the difficulty?

- Typical minesweeper configurations have many squares. (The so-called 'advanced' board has 99 mines in 480 squares, that's

  560220999337421345429058985775 8...95706311681983856732951596334816 00

  or over 10^{100} possible ways of placing these 99 mines.)

So what's the difficulty?

- Typical minesweeper configurations have many squares. (The so-called 'advanced' board has 99 mines in 480 squares, that's

  560220999337421345429058985775 8...95706311681983856732951596334816 00

  or over 10^{100} possible ways of placing these 99 mines.)
- No known method can search through all these possibilities in reasonable time.

Can we avoid this difficulty?

- Possibly. In fact no-one knows the answer to this question.

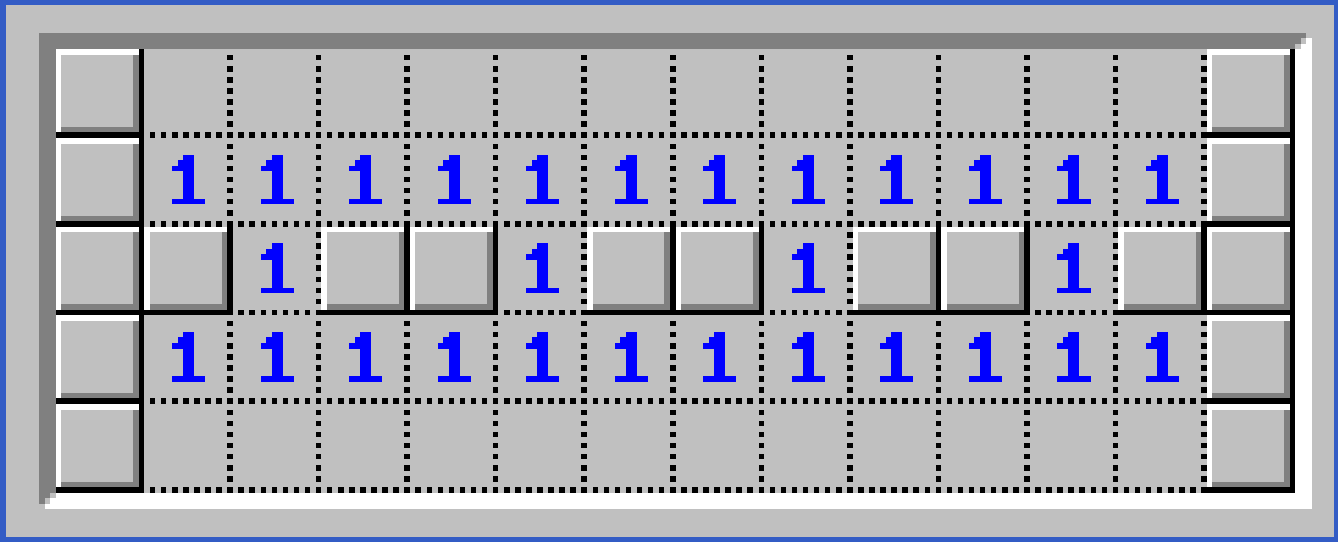
Can we avoid this difficulty?

- Possibly. In fact no-one knows the answer to this question.
- We can hope for a *fast* or *efficient* algorithm that takes time proportional to a fixed polynomial in the number of squares in the input configuration (*Polynomial Time*) rather than *Exponential Time*.

Can we avoid this difficulty?

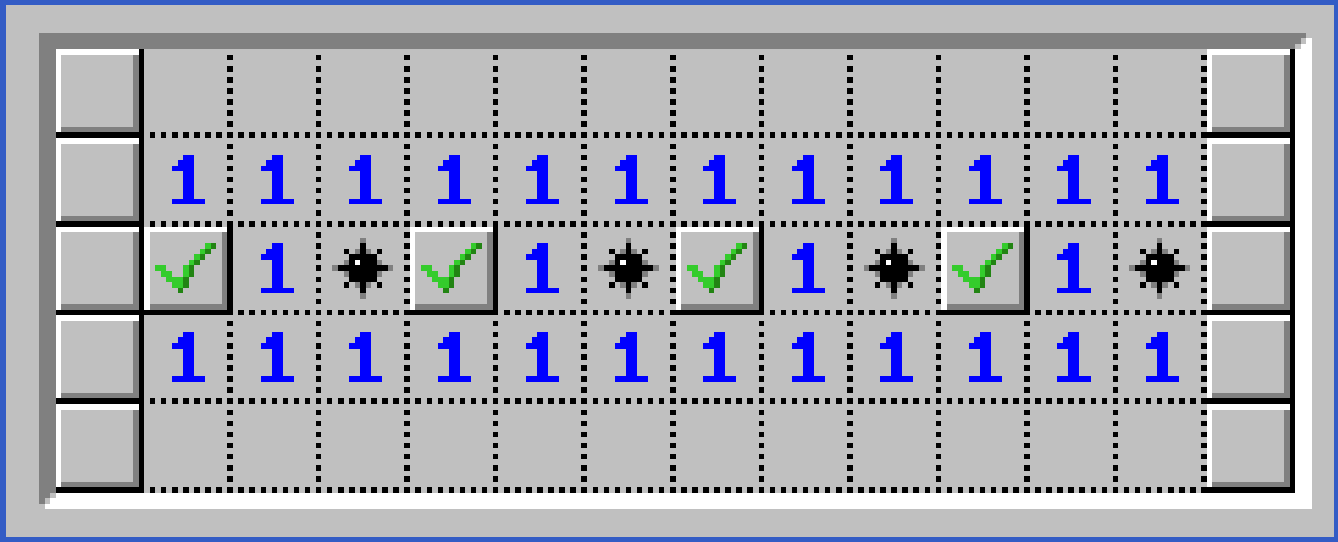
- Possibly. In fact no-one knows the answer to this question.
- We can hope for a *fast* or *efficient* algorithm that takes time proportional to a fixed polynomial in the number of squares in the input configuration (*Polynomial Time*) rather than *Exponential Time*.
- However, it seems unlikely that such an algorithm exists.

A wire



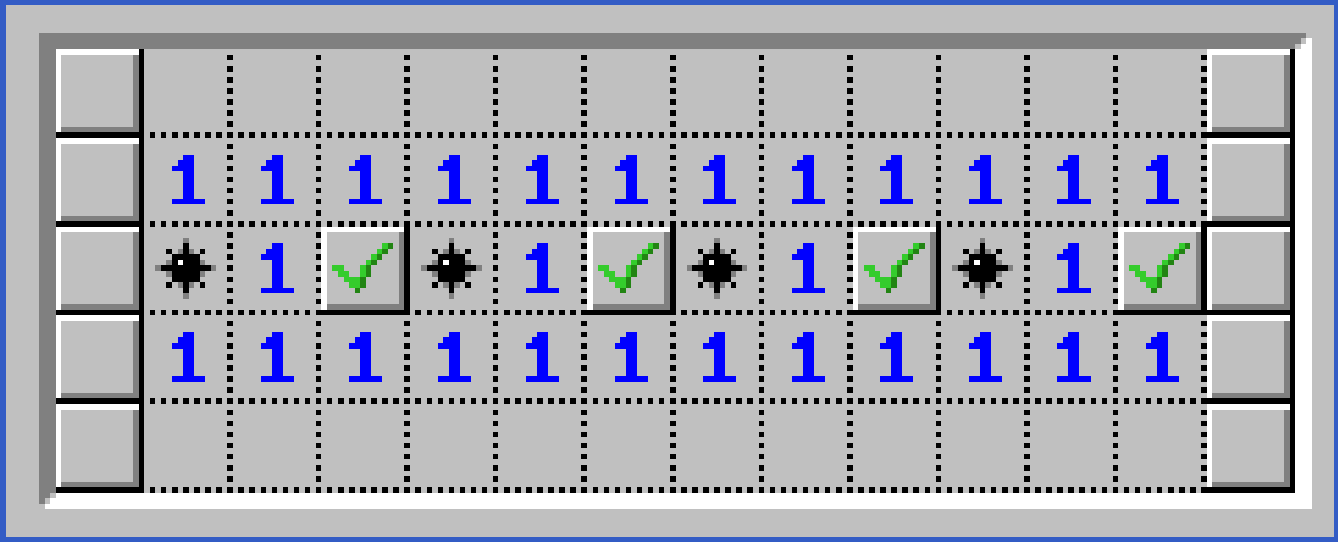
Minesweeper is complicated by the fact that something in one part of the board can affect the whole board.

A wire



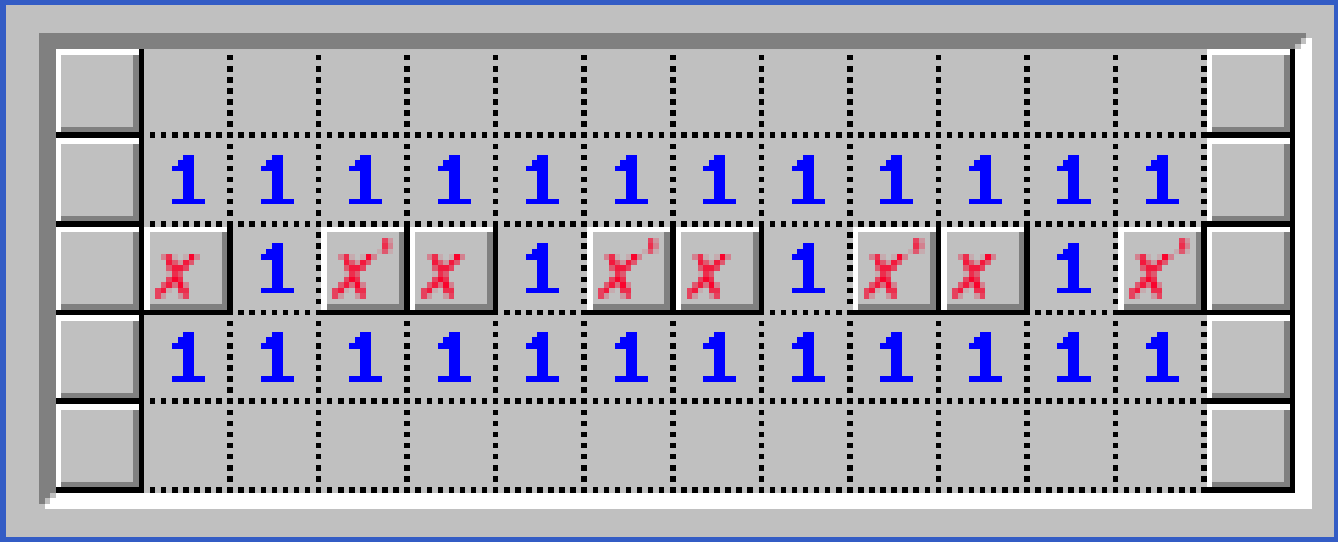
Minesweeper is complicated by the fact that something in one part of the board can affect the whole board.

A wire



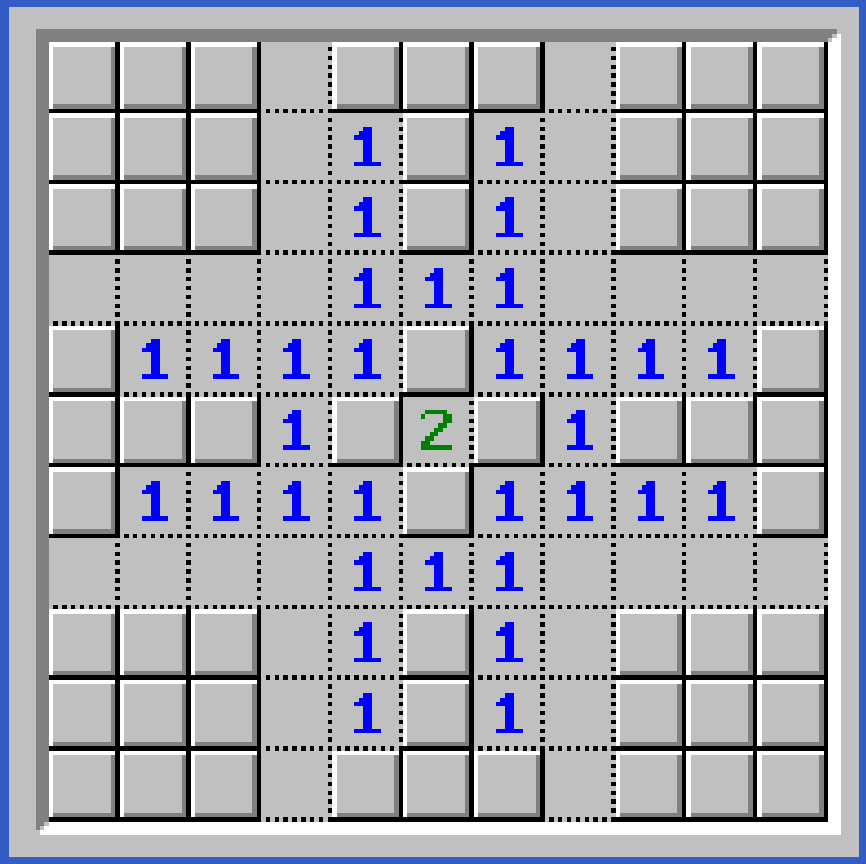
Minesweeper is complicated by the fact that something in one part of the board can affect the whole board.

A wire



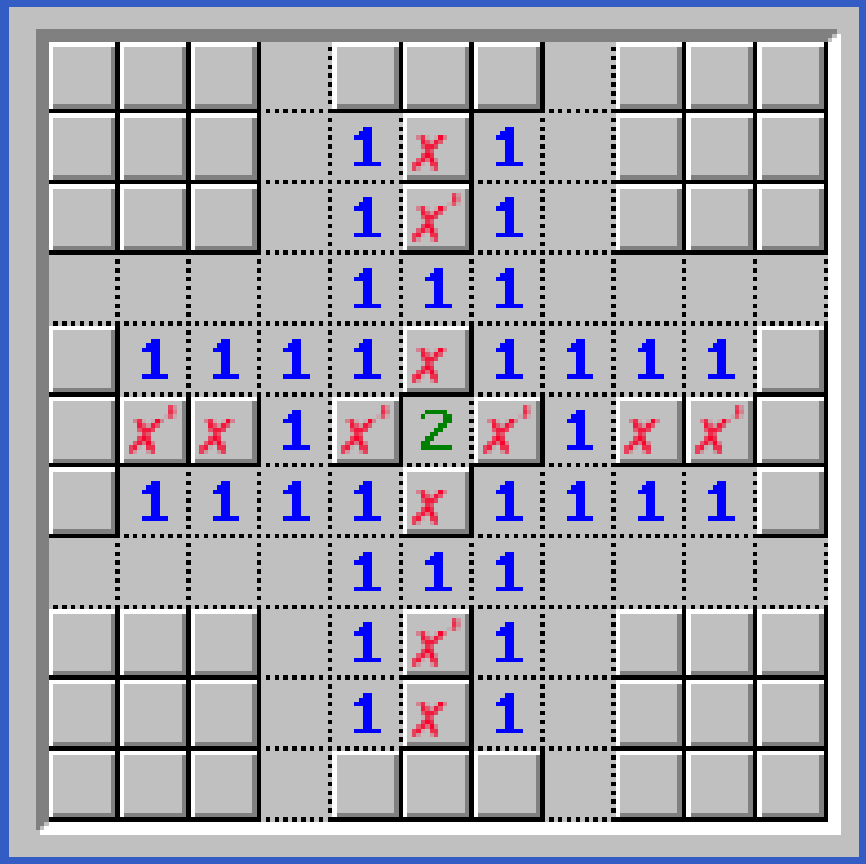
Minesweeper is complicated by the fact that something in one part of the board can affect the whole board.

A splitter



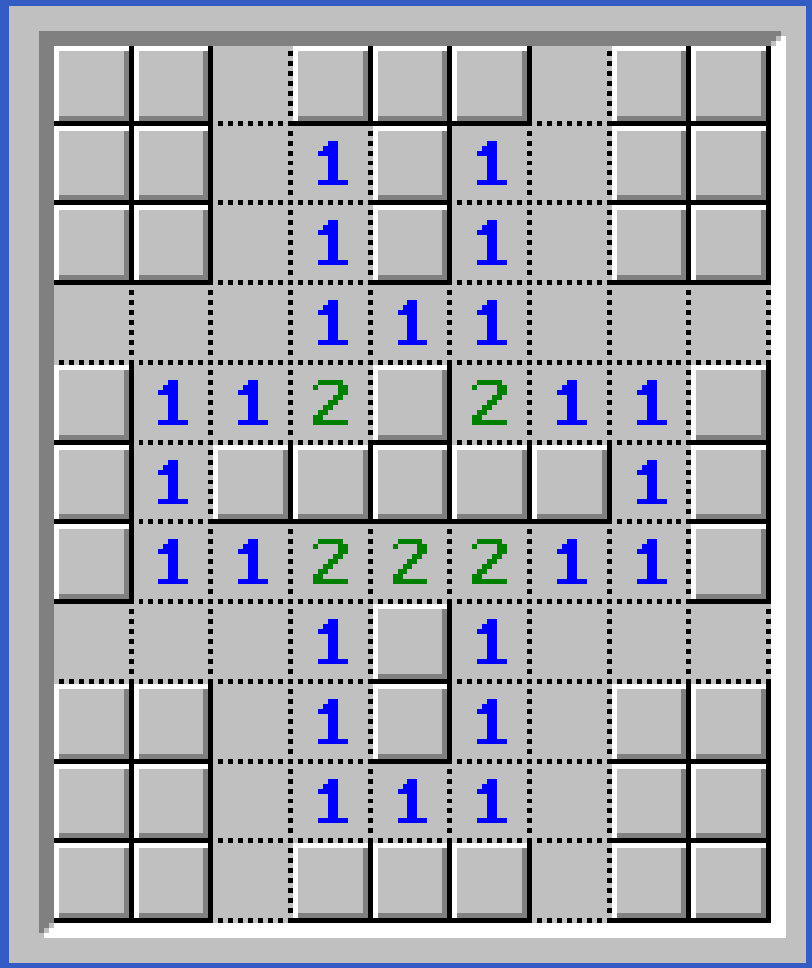
Use this to split, bend, or ‘invert’ wires.

A splitter



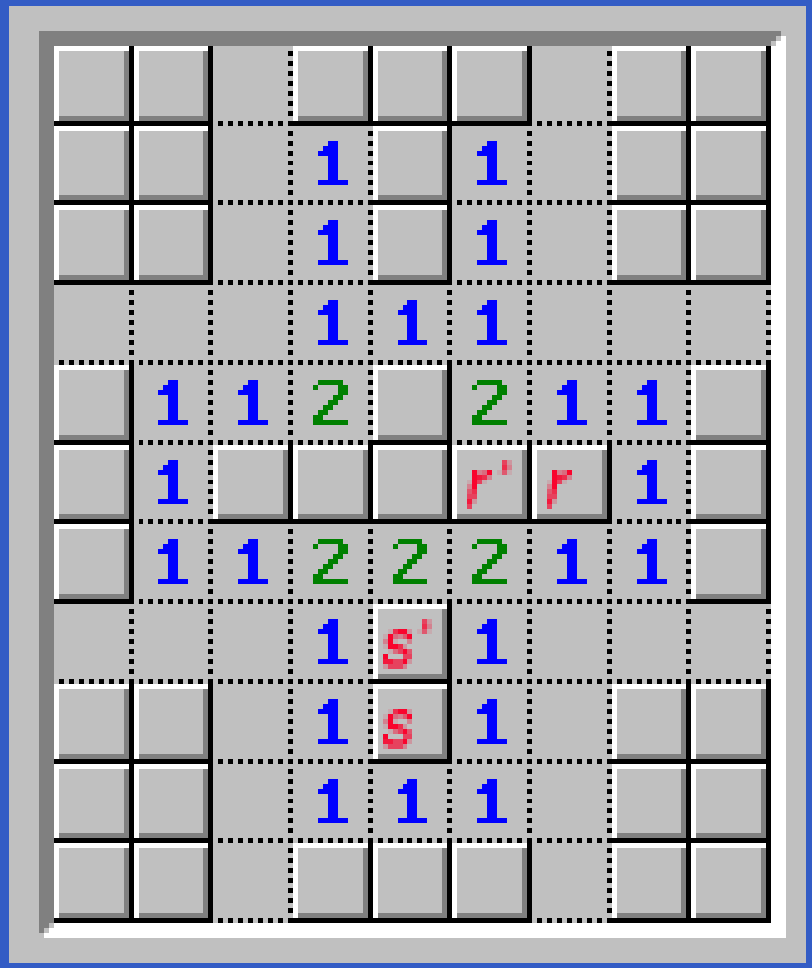
Use this to split, bend, or ‘invert’ wires.

A crossover



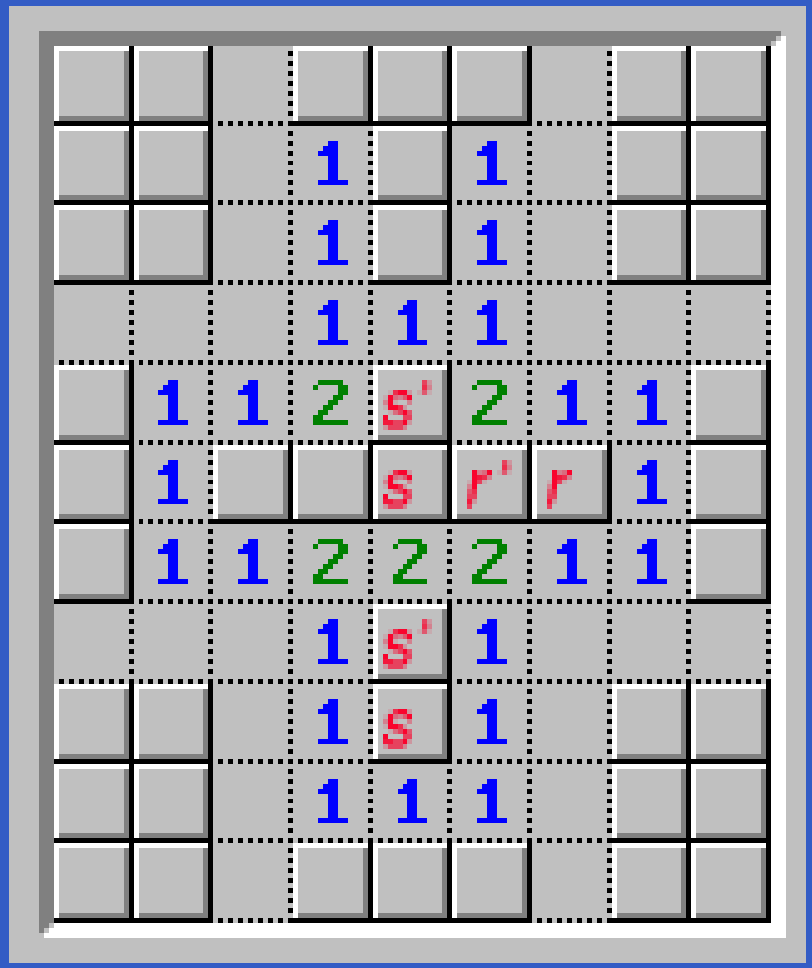
Use this to cross two wires over!

A crossover



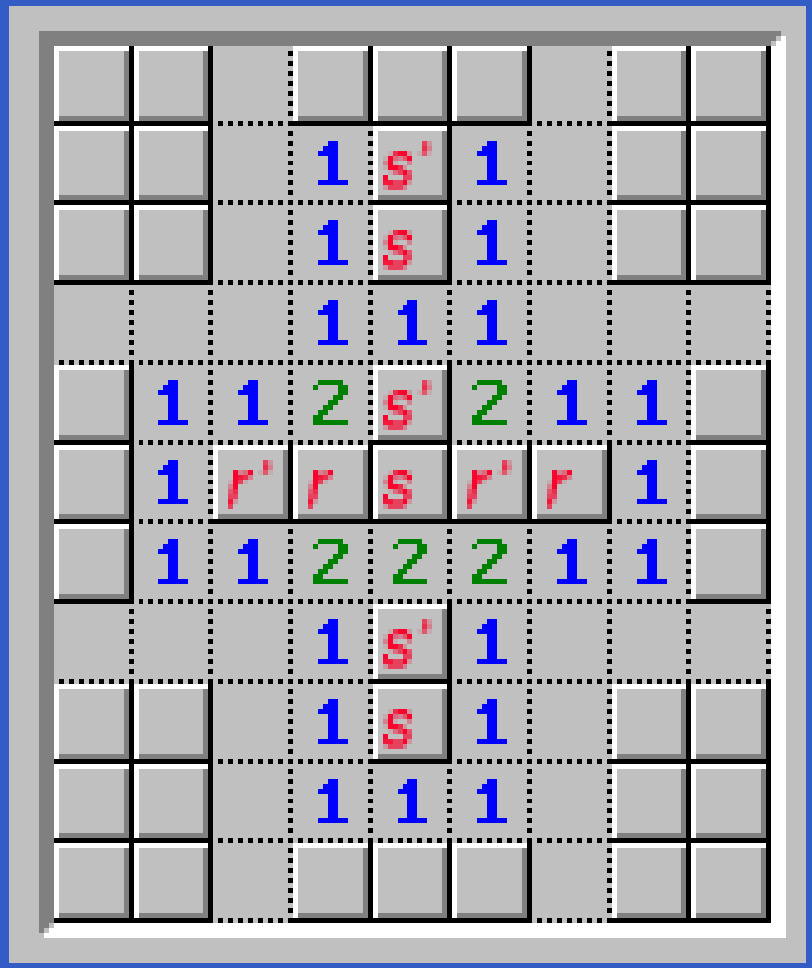
Use this to cross two wires over!

A crossover



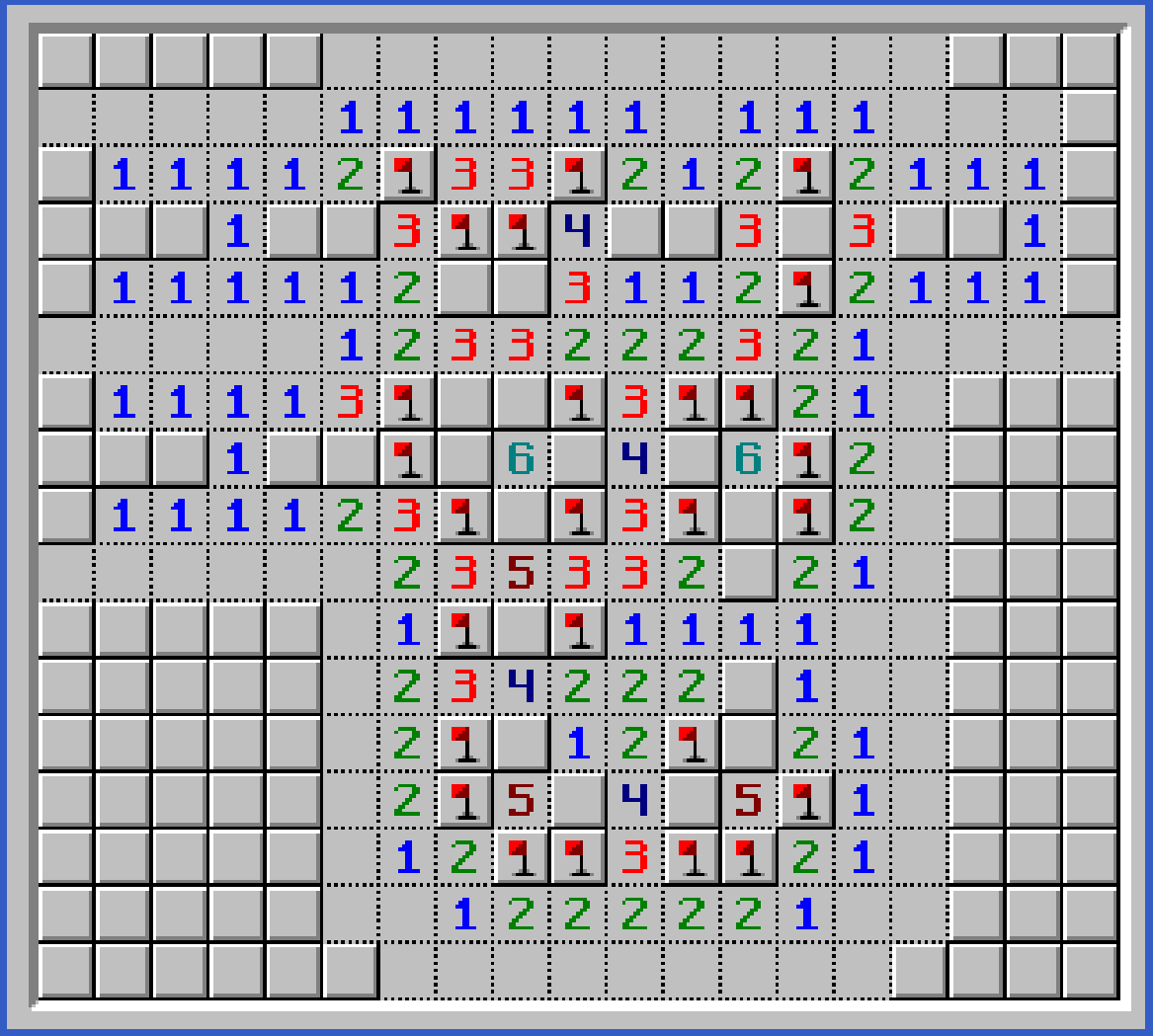
Use this to cross two wires over!

A crossover

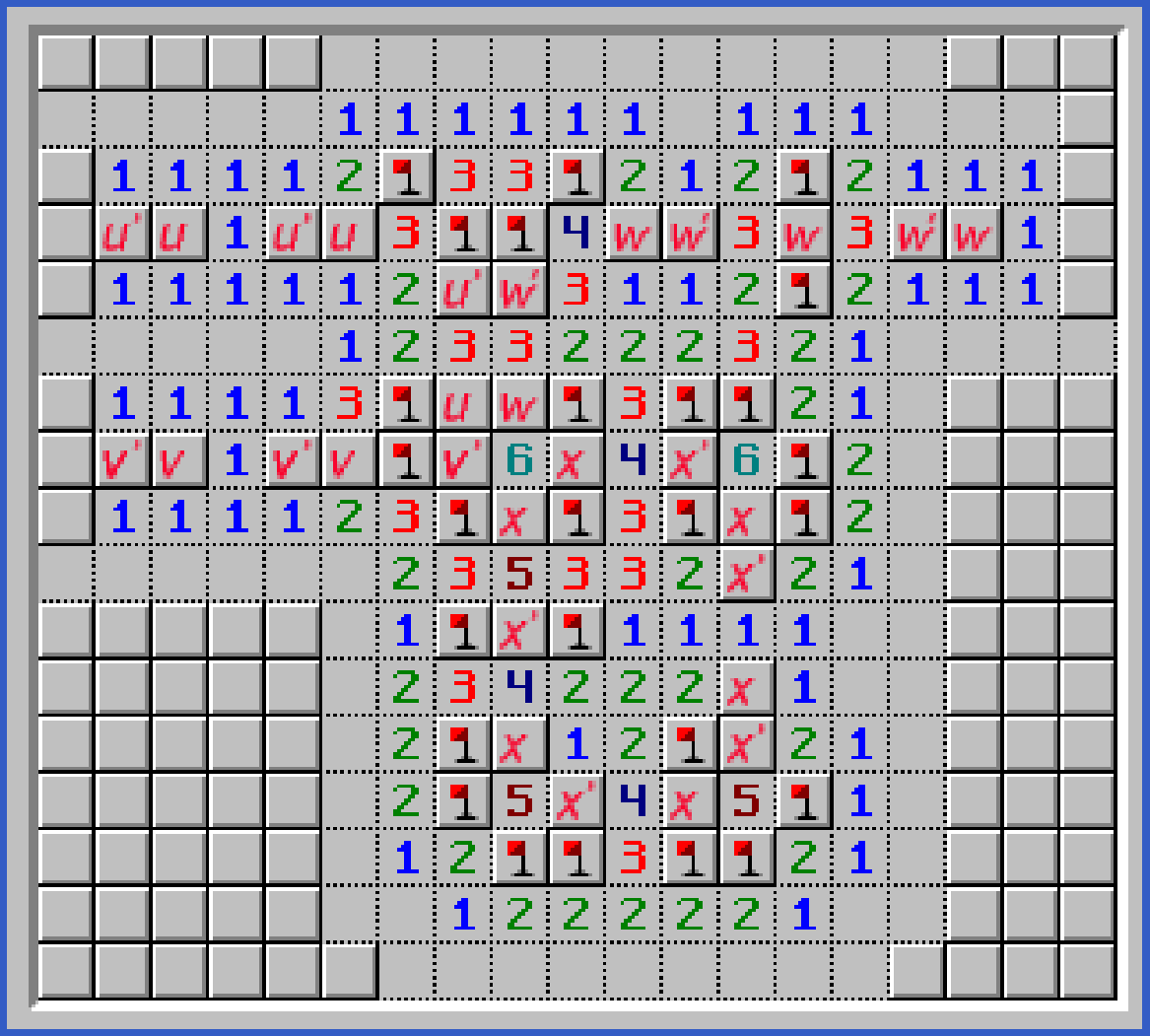


Use this to cross two wires over!

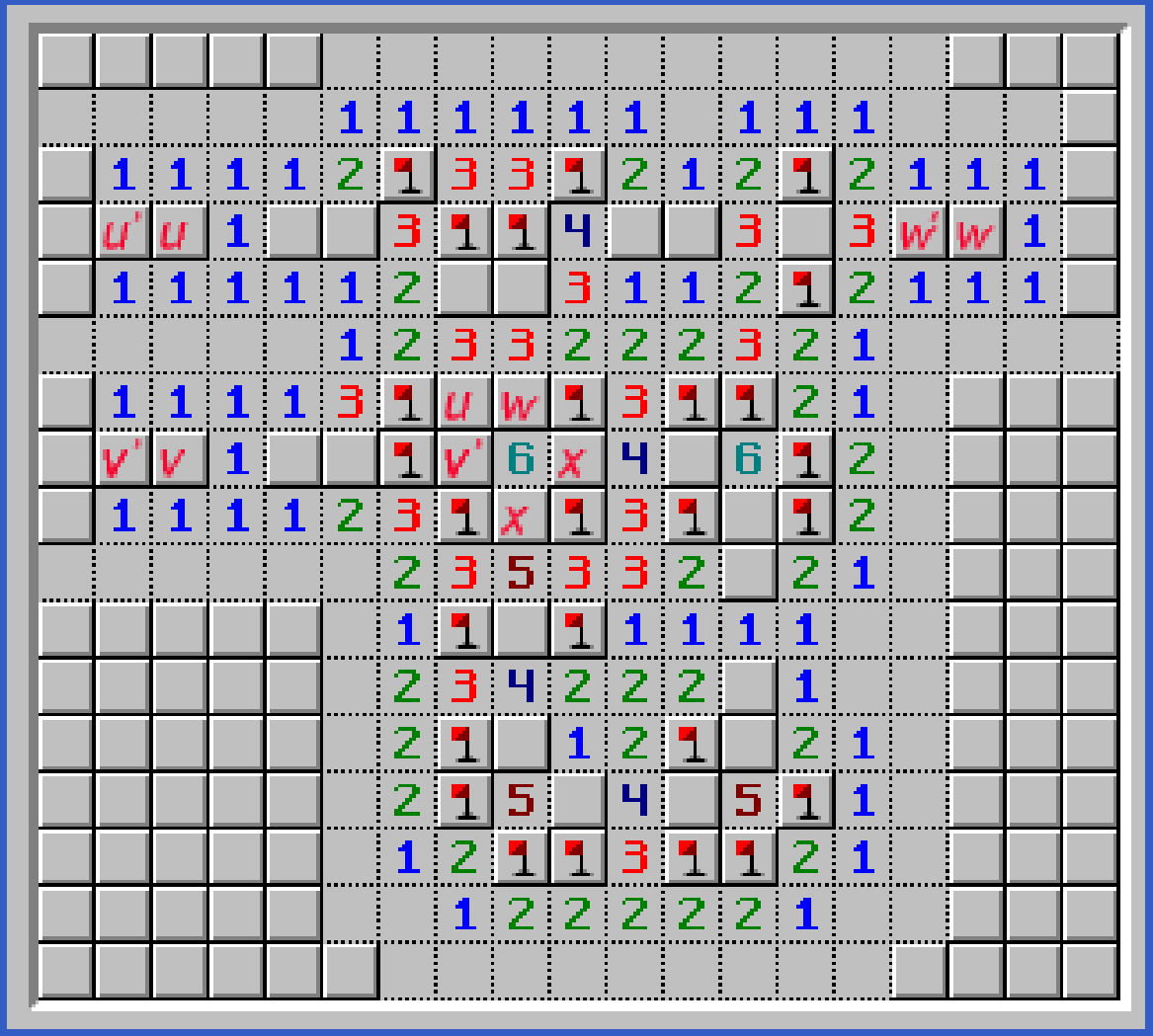
An XOR gate



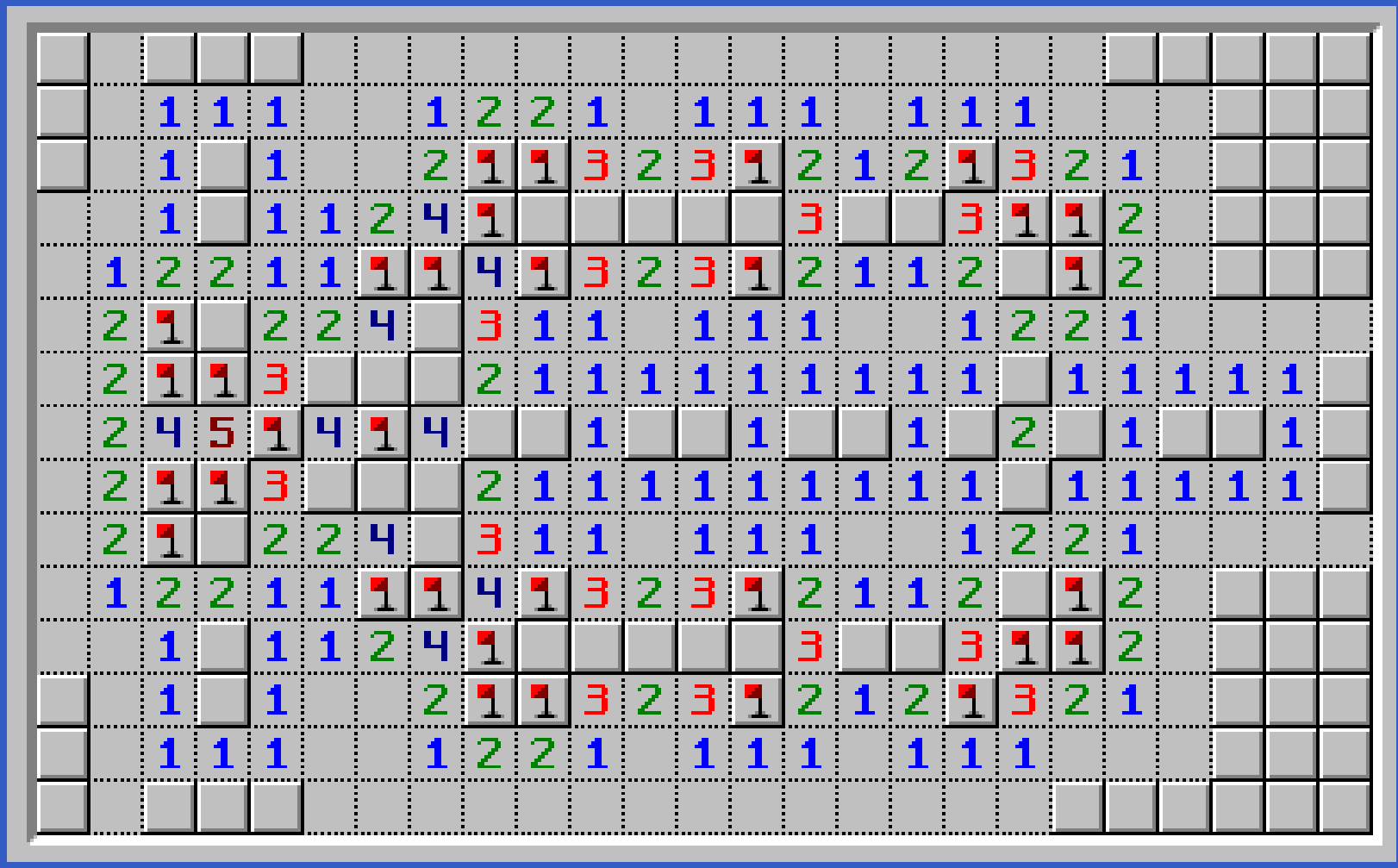
An XOR gate



An XOR gate

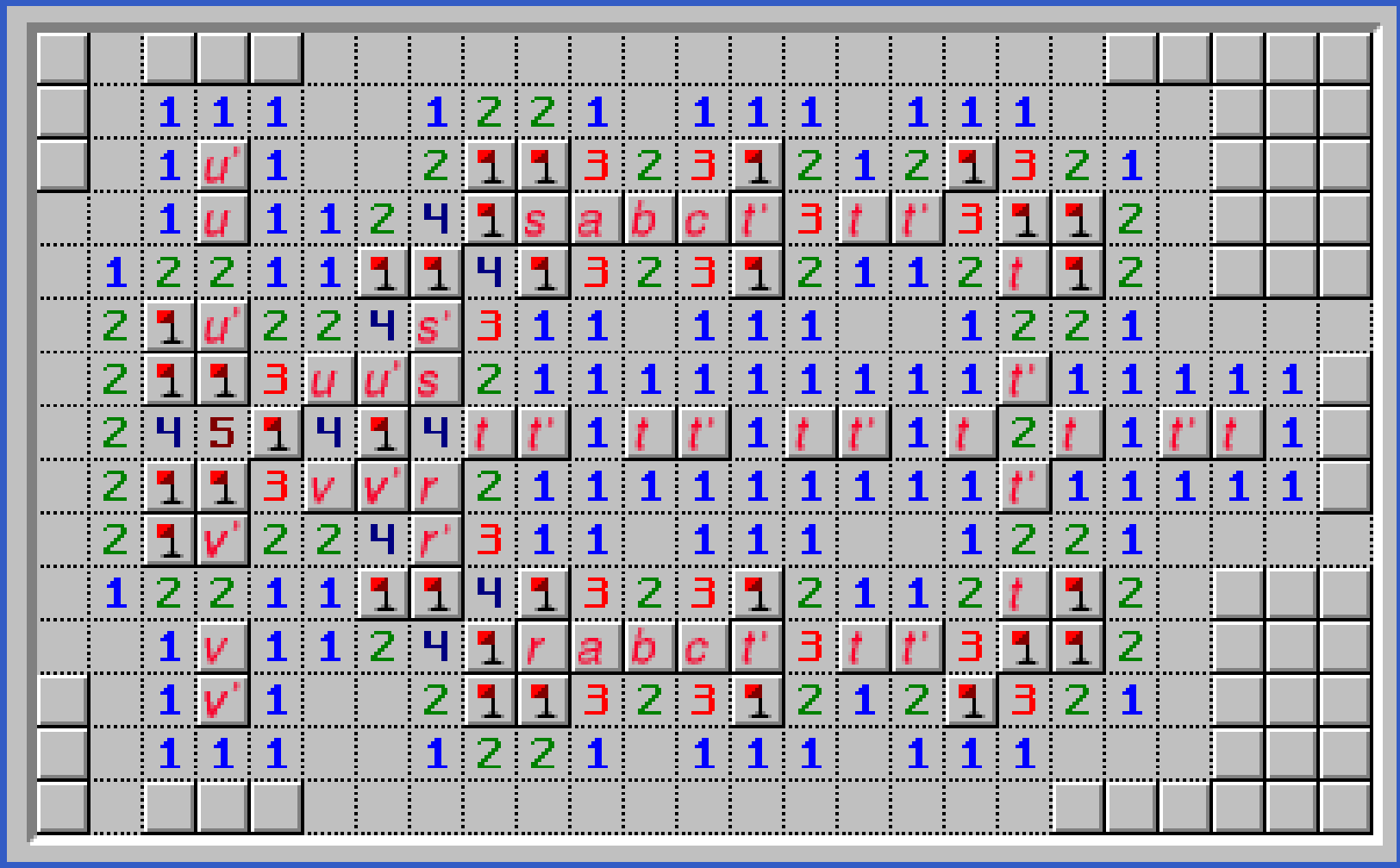


An AND gate



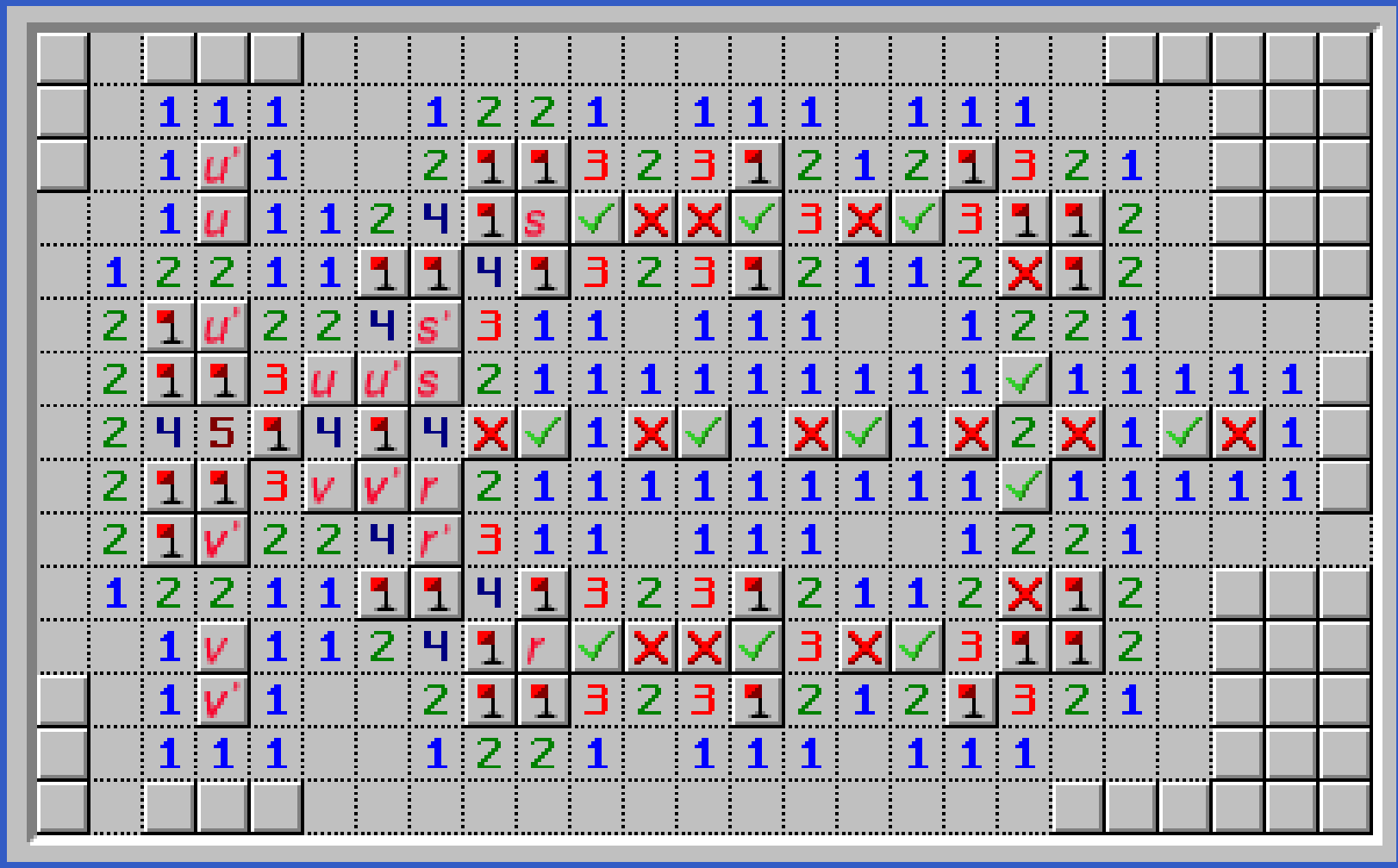
An AND gate. . .

An AND gate



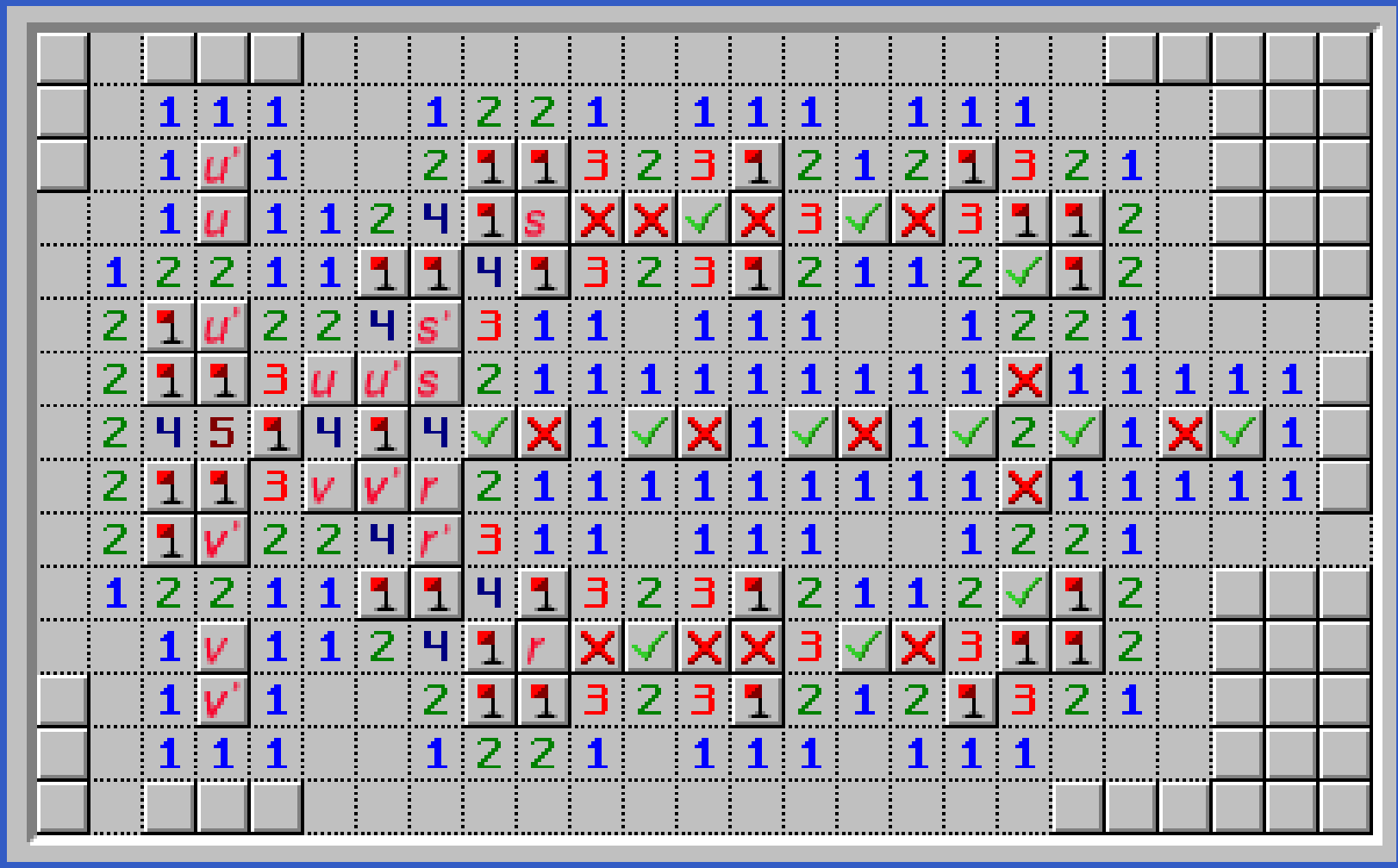
. . . and its internal wires.

An AND gate



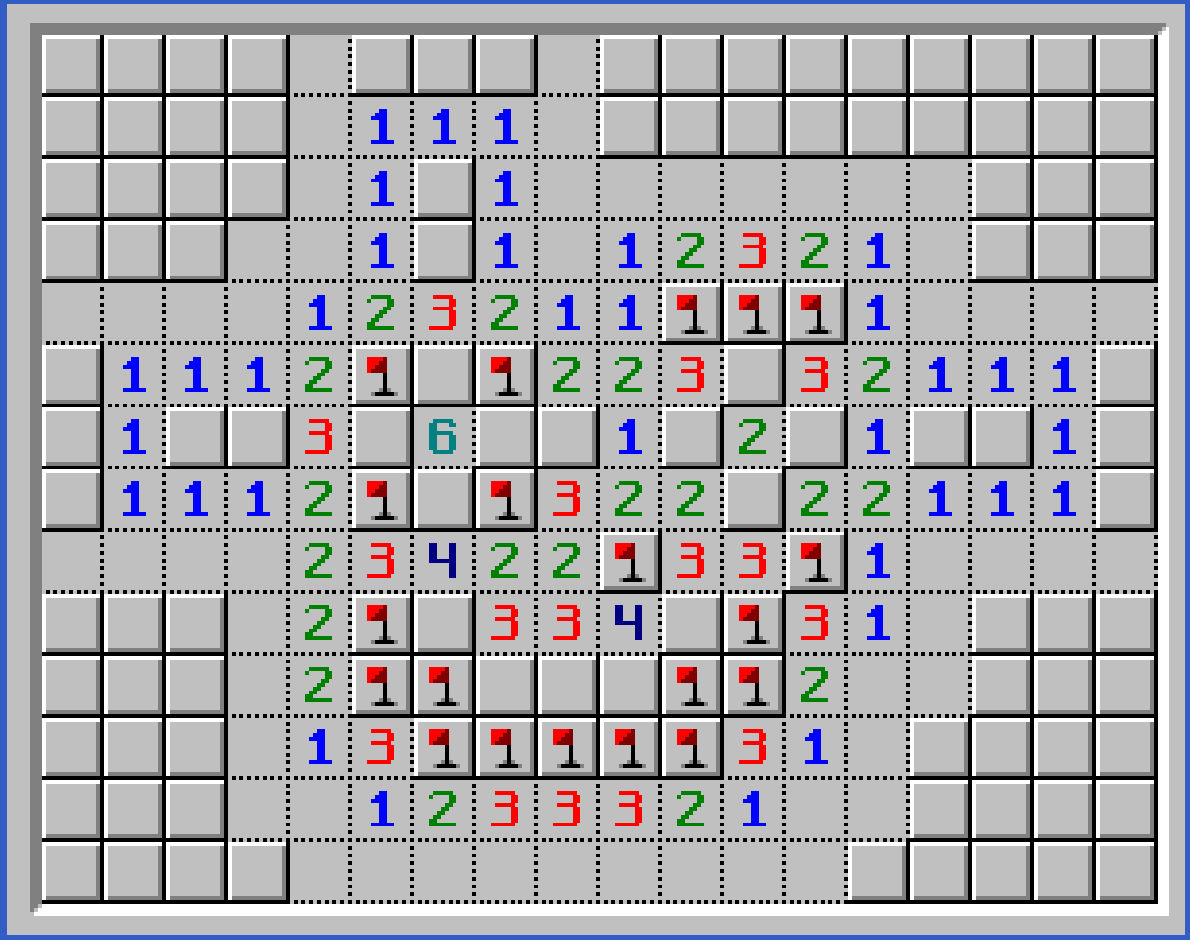
If *t* is a mine. . .

An AND gate



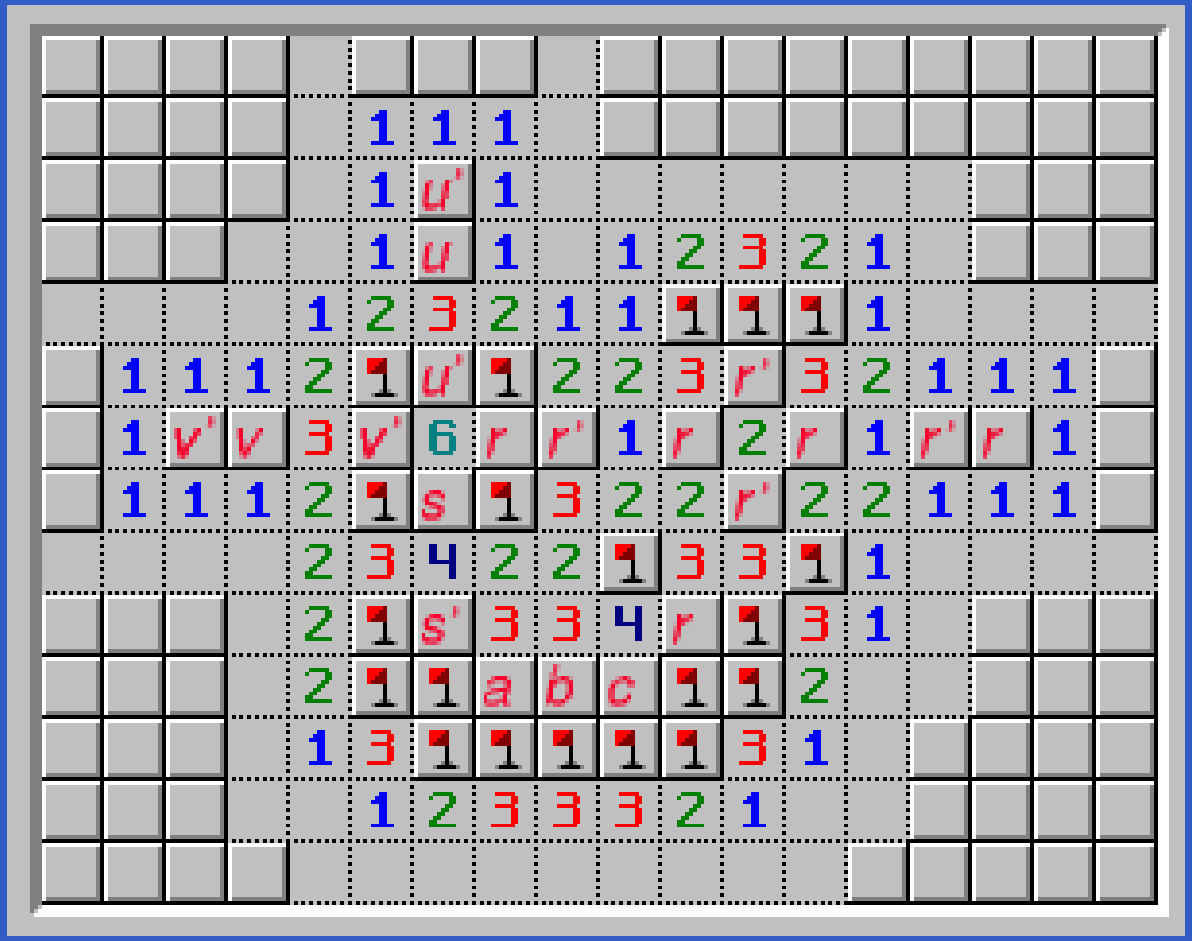
If *t* is not a mine. . .

An OR gate



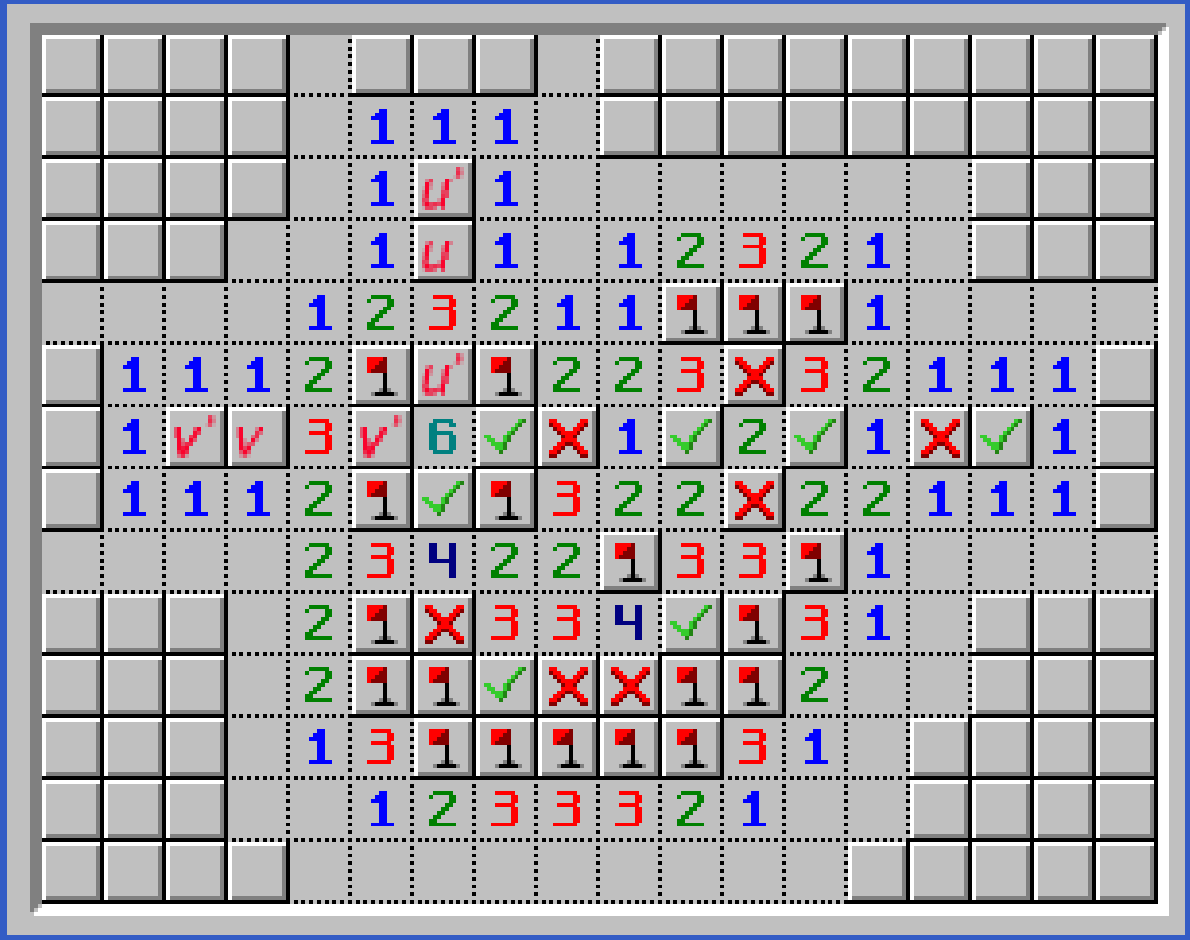
An OR gate. . .

An OR gate



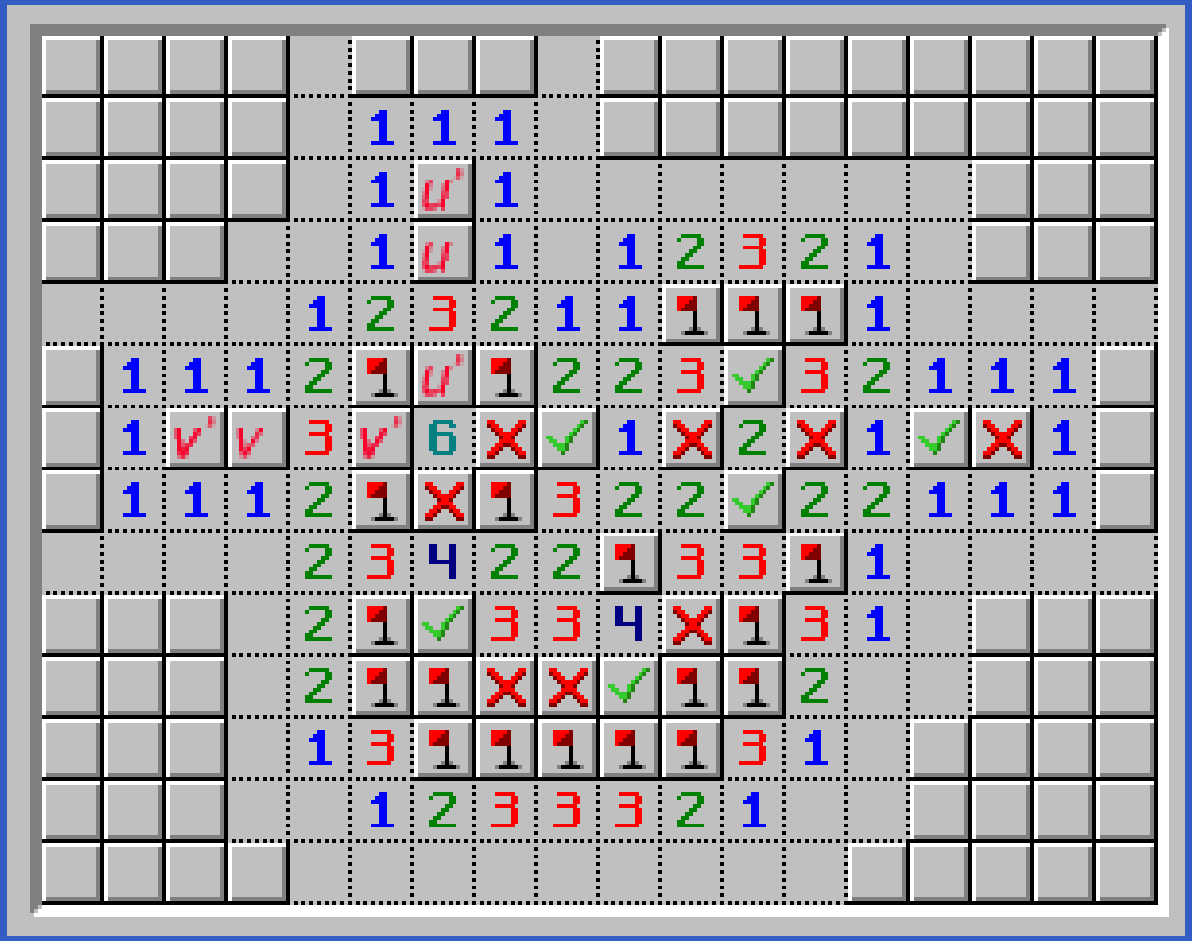
. . . with the internal wires shown.

An OR gate



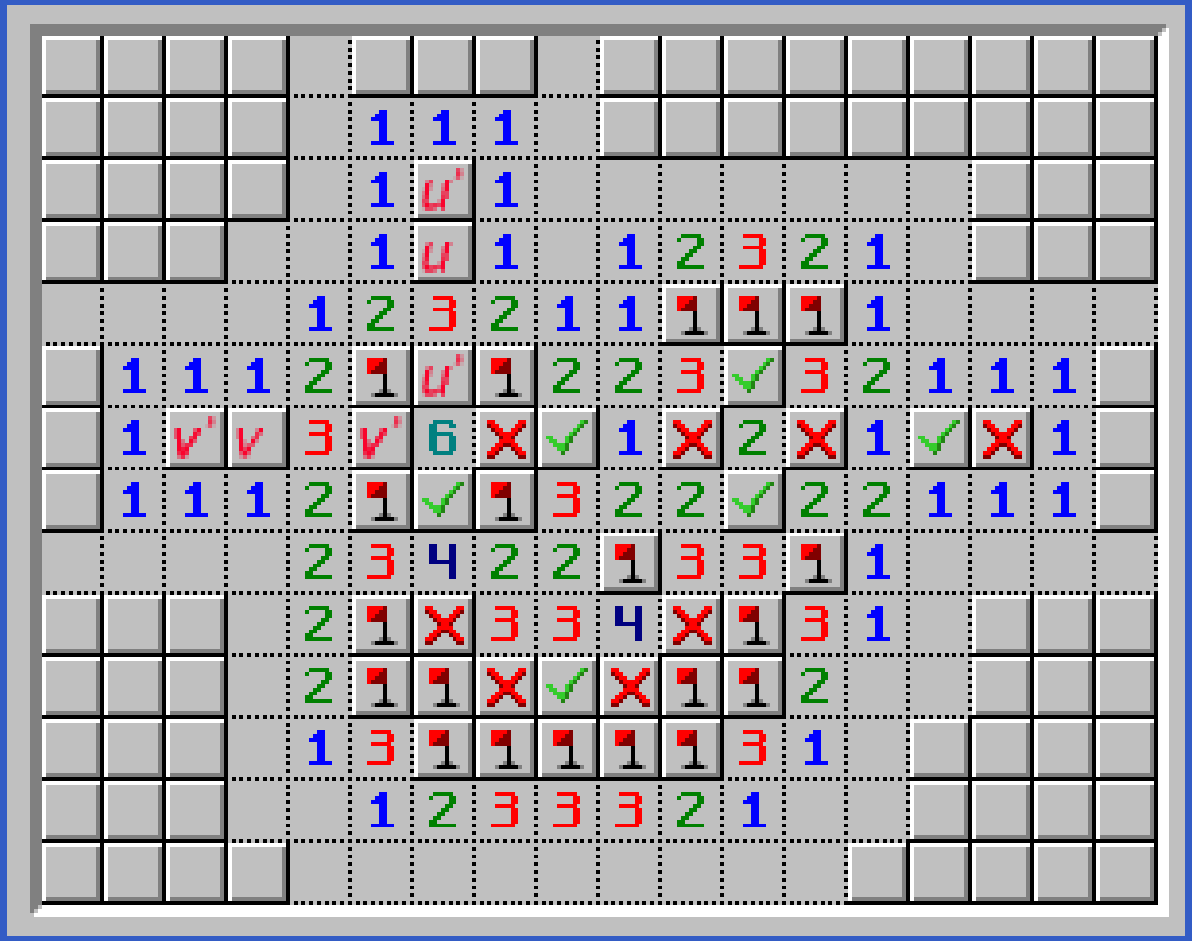
If *r* is free. . .

An OR gate



If *r* has a mine (case 1)...

An OR gate



If *r* has a mine (case 2)...

Minesweeper electronics

- Wire: allows us to carry signals around

Minesweeper electronics

- Wire: allows us to carry signals around
- Splitter: allows us to bend, split and invert signals in wires

Minesweeper electronics

- Wire: allows us to carry signals around
- Splitter: allows us to bend, split and invert signals in wires
- Crossover: allows us to make arbitrary circuits in the plain

Minesweeper electronics

- Wire: allows us to carry signals around
- Splitter: allows us to bend, split and invert signals in wires
- Crossover: allows us to make arbitrary circuits in the plain
- Logic gates: NOT AND OR XOR, etc.

NP-completeness

- We can build any logic circuits in minesweeper, so. . .

NP-completeness

- We can build any logic circuits in minesweeper, so. . .
- An algorithm solving the minesweeper consistency problem would allow us to solve questions about simple (propositional) logic.

NP-completeness

- We can build any logic circuits in minesweeper, so. . .
- An algorithm solving the minesweeper consistency problem would allow us to solve questions about simple (propositional) logic.
- Many other problems can be phrased as logic problems and therefore reduced to minesweeper consistency problems.

NP-completeness

- We can build any logic circuits in minesweeper, so. . .
- An algorithm solving the minesweeper consistency problem would allow us to solve questions about simple (propositional) logic.
- Many other problems can be phrased as logic problems and therefore reduced to minesweeper consistency problems.
- **The Minesweeper Consistency Problem is NP-complete.**

Some interesting NP problems

- The travelling salesman problem

Some interesting NP problems

- The travelling salesman problem
- Many other problems is scheduling, packing, decision maths, etc.

Some interesting NP problems

- The travelling salesman problem
- Many other problems is scheduling, packing, decision maths, etc.
- Other games, such as Tetris

Some interesting NP problems

- The travelling salesman problem
- Many other problems is scheduling, packing, decision maths, etc.
- Other games, such as Tetris
- Factorizing integers

Some interesting NP problems

- The travelling salesman problem
- Many other problems is scheduling, packing, decision maths, etc.
- Other games, such as Tetris
- Factorizing integers
- Determining a computer user's secret key from his public key

The P equals NP question

- The ‘P equals NP question’ asks if every NP problem has an efficient algorithm. No-one knows the answer! A prize of $1,000,000 has been offered for a proof either way.

The P equals NP question

- The ‘P equals NP question’ asks if every NP problem has an efficient algorithm. No-one knows the answer! A prize of $1,000,000 has been offered for a proof either way.
- It would suffice to show
  - *either* that the minesweeper consistency problem *has* and efficient solution
  - *or* that there is no such algorithm for the minesweeper consistency problem