```
Advent of Code [About] [Events] [Shop] [Settings] [Log Out] jhillierdavis 26*
     /^2023$/ [Calendar] [AoC++] [Sponsors] [Leaderboard] [Stats]
--- Day 12: Hot Springs ---
                                                                                make Advent of
You finally reach the hot springs! You can see steam rising from secluded
                                                                                Code possible:
areas attached to the primary, ornate building.
                                                                                 THE MERGE - The
As you turn to enter, the researcher stops you. "Wait - I thought you were
                                                                                Developer
looking for the hot springs, weren't you?" You indicate that this
                                                                                 Experience
definitely looks like hot springs to you.
                                                                                Conference in
                                                                                 Berlin, June 2024
"Oh, sorry, common mistake! This is actually the onsen! The hot springs are
                                                                                 (created by the
next door."
                                                                                 co-founder of
                                                                                GitHub)
You look in the direction the researcher is pointing and suddenly notice
the massive metal helixes towering overhead. "This way!"
It only takes you a few more steps to reach the main gate of the massive
fenced-off area containing the springs. You go through the gate and into a
small administrative building.
```

"Hello! What brings you to the hot springs today? Sorry they're not very

the missing machine parts for Desert Island.

enough to launch you up there!"

the spring is operational or damaged.

records.

this:

#.#.## 1,1,3

???.### 1,1,3

|.#...#...###. 1,1,3

####.#...#... 4,1,1

.###.##....# 3,2,1

.??...?##. 1,1,3

????.#...#... 4,1,1

?###???????? 3,2,1

criteria in each row.

?#?#?#?#?#?#? 1,3,1,6

????.######..#####. 1,6,5

1.#.###.#.##### 1,3,1,6

#...######..####. 1,6,5

hot right now; we're having a lava shortage at the moment." You ask about

"Oh, all of Gear Island is currently offline! Nothing is being manufactured

"Say, could you go up and see why the lava stopped flowing? The springs are

There's just one problem - many of the springs have fallen into disrepair,

yet, their condition records of which springs are damaged (your puzzle

In the giant field just outside, the springs are arranged into rows. For

operational (.) or damaged (#). This is the part of the condition records

However, the engineer that produced the condition records also duplicated

some of this information in a different format! After the list of springs

for a given row, the size of each contiguous group of damaged springs is

accounts for every damaged spring, and each number is the entire size of

So, condition records with no unknown spring conditions might look like

However, the condition records are partially damaged; some of the springs'

Equipped with this information, it is your job to figure out how many

different arrangements of operational and broken springs fit the given

In the first line (???.### 1,1,3), there is exactly one way separate groups

row: the first three unknown springs must be broken, then operational, then

of one, one, and three broken springs (in that order) can appear in that

its contiguous group (that is, groups are always separated by at least one

listed in the order those groups appear in the row. This list always

operational spring: #### would always be 4, never 2,2).

conditions are actually unknown (?). For example:

broken (#.#), making the whole row #.#.###.

- ????.######..####. 1,6,5 - 4 arrangements

- ?###???????? 3,2,1 - 10 arrangements

21 arrangements.

--- Part Two ---

Your puzzle answer was 7599.

counts?

by ,).

|.# 1|

So, this row:

Would become:

.#?.#?.#?.# 1,1,1,1,1

for some rows is now much larger:

- ???.### 1,1,3 - 1 arrangement

that is itself damaged; for some springs, it is simply unknown (?) whether

input) are also damaged! You'll need to help them repair the damaged

each row, the condition records show every spring and whether it is

so they're not actually sure which springs would even be safe to use! Worse

too cold for normal operation, but we should be able to find one springy

at the moment, not until we get more lava to heat our forges. And our

springs. The springs aren't very springy unless they're hot!"

The second line is more interesting: \(\begin{aligned} .??...?\frac{\pi}{2}....?\frac{\pi}{2}...?\frac{\pi}{2}...?\frac{\pi}{2}...?\frac{\pi}{ of four different arrangements. The last ? must always be broken (to satisfy the final contiguous group of three broken springs), and each ?? must hide exactly one of the two broken springs. (Neither ?? could be both broken springs or they would form a single contiguous group of two; if that were true, the numbers afterward would have been 2,3 instead.) Since each ?? can either be #. or .#, there are four possible arrangements of springs. The last line is actually consistent with ten different arrangements! Because the first number is 3, the first and second ? must both be  $\overline{.}$  (if either were #, the first number would have to be 4 or higher). However, the remaining run of unknown spring conditions have many different ways they could hold groups of two and one broken springs: ?###???????? 3,2,1 .###.##.#.... .###.##..#.. .###.##.... .###.##....# .###..##.. .###..##..#. .###..##..# .###...##.#. .###...# .###...##.# In this example, the number of possible arrangements for each row is: - ???.### 1,1,3 - 1 arrangement - .??...?##. 1,1,3 - 4 arrangements - ?#?#?#?#?#?#? 1,3,1,6 - 1 arrangement - ????.#...#... 4,1,1 - 1 arrangement

Adding all of the possible arrangement counts together produces a total of

For each row, count all of the different arrangements of operational and

As you look out at the field of springs, you feel like there are way more

discover that they were actually folded up this whole time!

The first line of the above example would become:

- <u>.</u>??...?##. 1,1,3 - 16384 arrangements

- ????.######..#####. 1,6,5 - 2500 arrangements

- ?#?#?#?#?#?#? 1,3,1,6 - 1 arrangement

- ?###???????? 3,2,1 - 506250 arrangements

- ????.#...#... 4,1,1 - 16 arrangements

springs than the condition records list. When you examine the records, you

To unfold the records, on each row, replace the list of spring conditions

contiguous groups of damaged springs with five copies of itself (separated

with five copies of itself (separated by ?) and replace the list of

???.###????.###????.###????.### 1,1,3,1,1,3,1,1,3,1,1,3,1,1,3

In the above example, after unfolding, the number of possible arrangements

broken springs that meet the given criteria. What is the sum of those

After unfolding, adding all of the possible arrangement counts together produces 525152. Unfold your condition records; what is the new sum of possible arrangement counts? Your puzzle answer was 15454556629917. Both parts of this puzzle are complete! They provide two gold stars: \*\* At this point, you should return to your Advent calendar and try another puzzle.

If you still want to see it, you can get your puzzle input.

You can also [Share] this puzzle.