

[Advent of Code 2022](#)
[\[About\]](#)
[\[Events\]](#)
[\[Shop\]](#)
[\[Settings\]](#)
[\[Log Out\]](#)
[Jarek 16★](#)  
[\[Calendar\]](#)
[\[AoC++\]](#)
[\[Sponsors\]](#)
[\[Leaderboard\]](#)
[\[Stats\]](#)

--- Day 9: Rope Bridge ---

This rope bridge creaks as you walk along it. You aren't sure how old it is, or whether it can even support your weight.

It seems to support the Elves just fine, though. The bridge spans a gorge which was carved out by the massive river far below you.

You step carefully; as you do, the ropes stretch and twist. You decide to distract yourself by modeling rope physics; maybe you can even figure out where not to step.

Consider a rope with a knot at each end; these knots mark the head and the tail of the rope. If the head moves far enough away from the tail, the tail is pulled toward the head.

Due to nebulous reasoning involving [Planck lengths](#), you should be able to model the positions of the knots on a two-dimensional grid. Then, by following a hypothetical series of motions (your puzzle input) for the head, you can determine how the tail will move.

Due to the aforementioned Planck lengths, the rope must be quite short; in fact, the head (**H**) and tail (**T**) must always be touching (diagonally adjacent and even overlapping both count as touching):

```
....
.TH.
....

....
.H.
..T.
....

...
.H. (H covers T)
...
```

If the head is ever two steps directly up, down, left, or right from the tail, the tail must also move one step in that direction so it remains close enough:

```
.....
.TH.. -> .T.H. -> ..TH.
.....

...    ...    ...
.T.    .T.    ...
.H. -> ... -> .T.
...    .H.    .H.
...    ...    ...
```

Otherwise, if the head and tail aren't touching and aren't in the same row or column, the tail always moves one step diagonally to keep up:

Our [sponsors](#) help make Advent of Code possible:

[TCGplayer](#) - Join an ever-growing worldwide team in connecting hobbyists to communities, exchanging things and thoughts that fuel passions. Competitive benefits, remote work, participate in hackathons and Advent of Code, & more!

```

.....   .....   .....
.....   ..H..   ..H..
..H..  ->  .....  ->  ..T..
.T...   .T...   .....
.....   .....   .....

.....   .....   .....
.....   .....   .....
..H..  ->  ...H.  ->  ..TH.
.T...   .T...   .....
.....   .....   .....

```

You just need to work out where the tail goes as the head follows a series of motions. Assume the head and the tail both start at the same position, overlapping.

For example:

```

R 4
U 4
L 3
D 1
R 4
D 1
L 5
R 2

```

This series of motions moves the head right four steps, then up four steps, then left three steps, then down one step, and so on. After each step, you'll need to update the position of the tail if the step means the head is no longer adjacent to the tail. Visually, these motions occur as follows (`S` marks the starting position as a reference point):

```

== Initial State ==

.....
.....
.....
.....
H..... (H covers T, s)

== R 4 ==

.....
.....
.....
.....
TH.... (T covers s)

.....
.....
.....
.....
sTH...

.....
.....
.....
.....
S.TH..

.....
.....
.....
.....
S..TH.

== U 4 ==

.....
.....
.....
...H.
S..T..

.....
.....
...H.
...T.
S.....

.....
...H.
...T.
.....
S.....

.....
...H.
...T.
.....
S.....

== L 3 ==

...H..
...T.

```

After simulating the rope, you can count up all of the positions the tail visited at least once. In this diagram, **S** again marks the starting position (which the tail also visited) and **#** marks other positions the tail visited:

```

::##::
...##.
.####.
....#.
S####.

```

So, there are **13** positions the tail visited at least once.

Simulate your complete hypothetical series of motions. How many positions does the tail of the rope visit at least once?

.....

S.....

To begin, get your puzzle input.

== D 1 ==

Answer:  [Submit]

..T...

You can also [Share] this puzzle.

.....

.....

S.....

.....

.....

== R 4 ==

.....

..T...

..H...

.....

.....

S.....

.....

.....

..T...

..H..

.....

.....

S.....

.....

.....

...TH.

.....

.....

S.....

.....

.....

....TH

.....

.....

S.....

.....

.....

== D 1 ==

.....

.....

....T.

....H

.....

S.....

.....

.....

== L 5 ==

.....

.....

....T.

....H.

.....

```
S.....  
  
.....  
....T..  
...H..  
.....  
S.....  
  
.....  
.....  
..HT..  
.....  
S.....  
  
.....  
.....  
.HT..  
.....  
S.....  
  
.....  
.....  
HT....  
.....  
S.....  
  
== R 2 ==  
  
.....  
.....  
.H.... (H covers T)  
.....  
S.....  
  
.....  
.....  
.TH..  
.....  
S.....
```