

Advent of Code [About] [Events] [Shop] [Settings] [Log Out] RodicaMihaelaVasilescu 42*
 12.2019 [Calendar] [AoC++] [Sponsors] [Leaderboard] [Stats]

--- Day 18: Many-Worlds Interpretation ---

As you approach Neptune, a planetary security system detects you and activates a giant tractor beam on Triton! You have no choice but to land.

A scan of the local area reveals only one interesting feature: a massive underground vault. You generate a map of the tunnels (your puzzle input). The tunnels are too narrow to move diagonally.

Only one entrance (marked @) is present among the open passages (marked .) and stone walls (#), but you also detect an assortment of keys (shown as lowercase letters) and doors (shown as uppercase letters). Keys of a given letter open the door of the same letter: a opens A, b opens B, and so on. You aren't sure which key you need to disable the tractor beam, so you'll need to collect all of them.

For example, suppose you have the following map:

```
#####
#b.A.@.a#
#####
```

Starting from the entrance (@), you can only access a large door (A) and a key (a). Moving toward the door doesn't help you, but you can move 2 steps to collect the key, unlocking A in the process:

```
#####
#b.....@#
#####
```

Then, you can move 6 steps to collect the only other key, b:

```
#####
#@.....#
#####
```

So, collecting every key took a total of 8 steps.

Here is a larger example:

```
#####
#f.D.E.e.C.b.A.@.a.B.c.#
#####
#d.....#
#####
```

The only reasonable move is to take key a and unlock door A:

```
#####
#f.D.E.e.C.b.....@.B.c.#
#####
#d.....#
#####
```

Then, do the same with key b:

```
#####
#f.D.E.e.C.@.....c.#
#####
#d.....#
#####
```

...and the same with key c:

Our sponsors help make Advent of Code possible:

Xebia - an international network of passionate technologists and craftspeople, dedicated to exploring and creating new frontiers in IT

```
#####
#f.D.E.e.....@.#
#####
#d.....#
#####
```

Now, you have a choice between keys `d` and `e`. While key `e` is closer, collecting it now would be slower in the long run than collecting key `d` first, so that's the best choice:

```
#####
#f...E.e.....#
#####
#@.....#
#####
```

Finally, collect key `e` to unlock door `E`, then collect key `f`, taking a grand total of `86` steps.

Here are a few more examples:

```
- #####
#.....b.C.D.f#
#.#####
#.....@.a.B.c.d.A.e.F.g#
#####
```

Shortest path is `132` steps: `b`, `a`, `c`, `d`, `f`, `e`, `g`

```
- #####
#i.G..c...e..H.p#
#####.#####
#j.A..b...f..D.o#
#####@#####
#k.E..a...g..B.n#
#####.#####
#l.F..d...h..C.m#
#####
```

Shortest paths are `136` steps;
one is: `a`, `f`, `b`, `j`, `g`, `n`, `h`, `d`, `l`, `o`, `e`, `p`, `c`, `i`, `k`, `m`

```
- #####
#@.....ac.GI.b#
###d#e#f#####
###A#B#C#####
###g#h#i#####
#####
```

Shortest paths are `81` steps; one is: `a`, `c`, `f`, `i`, `d`, `g`, `b`, `e`, `h`

How many steps is the shortest path that collects all of the keys?

Your puzzle answer was `5450`.

The first half of this puzzle is complete! It provides one gold star: *

--- Part Two ---

You arrive at the vault only to discover that there is not one vault, but four - each with its own entrance.

On your map, find the area in the middle that looks like this:

```
...
.@.
...
```

Update your map to instead use the correct data:

```
@#@
###
@#@
```

This change will split your map into four separate sections, each with its own entrance:

```
#####
#a.#Cd#    #a.#Cd#
##...##    ##@#@##
##.@.##    --> #####
##...##    ##@#@##
#cB#Ab#    #cB#Ab#
#####
```

Because some of the keys are for doors in other vaults, it would take much too long to collect all of the keys by yourself. Instead, you deploy four remote-controlled robots. Each starts at one of the entrances (@).

Your goal is still to collect all of the keys in the fewest steps, but now, each robot has its own position and can move independently. You can only remotely control a single robot at a time. Collecting a key instantly unlocks any corresponding doors, regardless of the vault in which the key or door is found.

For example, in the map above, the top-left robot first collects key **a**, unlocking door **A** in the bottom-right vault:

```
#####
#@.#Cd#
##.@##
#####
##@#@##
#cB#.b#
#####
```

Then, the bottom-right robot collects key **b**, unlocking door **B** in the bottom-left vault:

```
#####
#@.#Cd#
##.@##
#####
##@#.##
#c.#.@#
#####
```

Then, the bottom-left robot collects key **c**:

```
#####
#@.#.d#
##.@##
#####
##.##
#@.#.@#
#####
```

Finally, the top-right robot collects key **d**:

```
#####
#@.#.@#
##.##
#####
##.##
#@.#.@#
#####
```

In this example, it only took **8** steps to collect all of the keys.

Sometimes, multiple robots might have keys available, or a robot might have to wait for multiple keys to be collected:

```
#####
#d.ABC.#.....a#
#####@#@#####
#####
#####@#@#####
#b.....#.....c#
#####
```

First, the top-right, bottom-left, and bottom-right robots take turns collecting keys **a**, **b**, and **c**, a total of $6 + 6 + 6 = 18$ steps. Then, the top-left robot can access key **d**, spending another **6** steps; collecting all of the keys here takes a minimum of **24** steps.

Here's a more complex example:

```
#####
#DcBa.#.GhKl#
#...#@#@I###
#e#d#####j#k#
###C#@#@###J#
#fEbA.#.FgHi#
#####
```

- Top-left robot collects key **a**.
- Bottom-left robot collects key **b**.
- Top-left robot collects key **c**.
- Bottom-left robot collects key **d**.
- Top-left robot collects key **e**.
- Bottom-left robot collects key **f**.
- Bottom-right robot collects key **g**.
- Top-right robot collects key **h**.
- Bottom-right robot collects key **i**.
- Top-right robot collects key **j**.
- Bottom-right robot collects key **k**.
- Top-right robot collects key **l**.

In the above example, the fewest steps to collect all of the keys is **32**.

Here's an example with more choices:

```
#####
#g#f.D#..h#l#
#F###e#E###.#
#dCba#@@BcIJ#
#####
#nK.L#@#G...#
#M###N#H###.#
#o#m..#i#j#k.#
#####
```

One solution with the fewest steps is:

- Top-left robot collects key **e**.

- Top-right robot collects key `h`.
- Bottom-right robot collects key `i`.
- Top-left robot collects key `a`.
- Top-left robot collects key `b`.
- Top-right robot collects key `c`.
- Top-left robot collects key `d`.
- Top-left robot collects key `f`.
- Top-left robot collects key `g`.
- Bottom-right robot collects key `k`.
- Bottom-right robot collects key `j`.
- Top-right robot collects key `l`.
- Bottom-left robot collects key `n`.
- Bottom-left robot collects key `m`.
- Bottom-left robot collects key `o`.

This example requires at least `72` steps to collect all keys.

After updating your map and using the remote-controlled robots, what is the fewest steps necessary to collect all of the keys?

Although it hasn't changed, you can still [get your puzzle input](#).

Answer: [\[Submit\]](#)

You can also [\[Share\]](#) this puzzle.