

Advent of Code [About] [Events] [Shop] [Settings] [Log Out] tam 27*
 <y>2023</y> [Calendar] [AoC++] [Sponsors] [Leaderboard] [Stats]

--- Day 25: Snowverload ---

Still somehow without snow, you go to the last place you haven't checked: the center of Snow Island, directly below the waterfall.

Here, someone has clearly been trying to fix the problem. Scattered everywhere are hundreds of weather machines, almanacs, communication modules, hoof prints, machine parts, mirrors, lenses, and so on.

Somehow, everything has been wired together into a massive snow-producing apparatus, but nothing seems to be running. You check a tiny screen on one of the communication modules: `Error 2023`. It doesn't say what `Error 2023` means, but it does have the phone number for a support line printed on it.

"Hi, you've reached Weather Machines And So On, Inc. How can I help you?" You explain the situation.

"Error 2023, you say? Why, that's a power overload error, of course! It means you have too many components plugged in. Try unplugging some components and--" You explain that there are hundreds of components here and you're in a bit of a hurry.

"Well, let's see how bad it is; do you see a big red reset button somewhere? It should be on its own module. If you push it, it probably won't fix anything, but it'll report how overloaded things are." After a minute or two, you find the reset button; it's so big that it takes two hands just to get enough leverage to push it. Its screen then displays:

```
SYSTEM OVERLOAD!

Connected components would require
power equal to at least 100 stars!
```

"Wait, how many components did you say are plugged in? With that much equipment, you could produce snow for an entire--" You disconnect the call.

You have nowhere near that many stars - you need to find a way to disconnect at least half of the equipment here, but it's already Christmas! You only have time to disconnect three wires.

Fortunately, someone left a wiring diagram (your puzzle input) that shows how the components are connected. For example:

```
jqt: rhn xhk nvd
rsh: frs pzl lsr
xhk: hfx
cmg: qnr nvd lhk bvb
rhn: xhk bvb hfx
bvb: xhk hfx
pzl: lsr hfx nvd
qnr: nvd
ntq: jqt hfx bvb xhk
nvd: lhk
lsr: lhk
rzs: qnr cmg lsr rsh
frs: qnr lhk lsr
```

Each line shows the name of a component, a colon, and then a list of other components to which that component is connected. Connections aren't directional; `abc: xyz` and `xyz: abc` both represent the same configuration. Each connection between two components is represented only once, so some components might only ever appear on the left or right side of a colon.

In this example, if you disconnect the wire between `hfx/pzl`, the wire between `bvb/cmg`, and the wire between `nvd/jqt`, you will divide the components into two separate, disconnected groups:

Our sponsors help make Advent of Code possible:

[Boot.dev](#) - React to become a backend developer? If you like AoC, you might be like us. We think smartest way to learn to code is to ensure you're never bored. Try the most captivating addictive way to learn to code on Boot.dev.

- 9 components: `cmg`, `frs`, `lhk`, `lsr`, `nvd`, `pzl`, `qnr`, `rsh`, and `rzs`.
- 6 components: `bvb`, `hfx`, `jqt`, `ntq`, `rhn`, and `xhk`.

Multiplying the sizes of these groups together produces `54`.

Find the three wires you need to disconnect in order to divide the components into two separate groups. What do you get if you multiply the sizes of these two groups together?

To begin, [get your puzzle input](#).

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

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