--- Day 19: Aplenty ---

The Elves of Gear Island are thankful for your help and send you on your way. They even have a hang glider that someone stole from Desert Island; since you're already going that direction, it would help them a lot if you would use it to get down there and return it to them.

As you reach the bottom of the relentless avalanche of machine parts, you discover that they're already forming a formidable heap. Don't worry, though - a group of Elves is already here organizing the parts, and they have a system.

To start, each part is rated in each of four categories:

- `x`: Extremely cool looking
- `m`: Musical (it makes a noise when you hit it)
- `a`: Aerodynamic
- `s`: Shiny

Then, each part is sent through a series of workflows that will ultimately accept or reject the part. Each workflow has a name and contains a list of rules; each rule specifies a condition and where to send the part if the condition is true. The first rule that matches the part being considered is applied immediately, and the part moves on to the destination described by the rule. (The last rule in each workflow has no condition and always applies if reached.)

Consider the workflow `ex{x>10:one,m<20:two,a>30:R,A}`. This workflow is named `ex` and contains four rules. If workflow `ex` were considering a specific part, it would perform the following steps in order:

- Rule "`x>10:one`": If the part's `x` is more than `10`, send the part to the workflow named `one`.
- Rule "`m<20:two`": Otherwise, if the part's `m` is less than `20`, send the part to the workflow named `two`.
- Rule "`a>30:R`": Otherwise, if the part's `a` is more than `30`, the part is immediately rejected (`R`).
- Rule "`A`": Otherwise, because no other rules matched the part, the part is immediately accepted (`A`).

If a part is sent to another workflow, it immediately switches to the start of that workflow instead and never returns. If a part is accepted (sent to `A`) or rejected (sent to `R`), the part immediately stops any further processing.

The system works, but it's not keeping up with the torrent of weird metal shapes. The Elves ask if you can help sort a few parts and give you the list of workflows and some part ratings (your puzzle input). For example:

```
px{a<2006:qkq,m>2090:A,rfg}
pv{a>1716:R,A}
lnx{m>1548:A,A}
rfg{s<537:gd,x>2440:R,A}
qs{s>3448:A,lnx}
qkq{x<1416:A,crn}
crn{x>2662:A,R}
in{s<1351:px,qqz}
qqz{s>2770:qs,m<1801:hdj,R}
gd{a>3333:R,R}
hdj{m>838:A,pv}

{x=787,m=2655,a=1222,s=2876}
{x=1679,m=44,a=2067,s=496}
{x=2036,m=264,a=79,s=2244}
{x=2461,m=1339,a=466,s=291}
{x=2127,m=1623,a=2188,s=1013}
```

The workflows are listed first, followed by a blank line, then the ratings of the parts the Elves would like you to sort. All parts begin in the workflow named `in`. In this example, the five listed parts go through the following workflows:

- `{x=787,m=2655,a=1222,s=2876}`: `in` -> `qqz` -> `qs` -> `lnx` -> `A`
- `{x=1679,m=44,a=2067,s=496}`: `in` -> `px` -> `rfg` -> `gd` -> `R`
- `{x=2036,m=264,a=79,s=2244}`: `in` -> `qqz` -> `hdj` -> `pv` -> `A`
- `{x=2461,m=1339,a=466,s=291}`: `in` -> `px` -> `qkq` -> `crn` -> `R`
- `{x=2127,m=1623,a=2188,s=1013}`: `in` -> `px` -> `rfg` -> `A`

Ultimately, three parts are accepted. Adding up the `x`, `m`, `a`, and `s` rating for each of the accepted parts gives `7540` for the part with `x=787`, `4623` for the part with `x=2036`, and `6951` for the part with `x=2127`. Adding all of the ratings for all of the accepted parts gives the sum total of `19114`.

Sort through all of the parts you've been given; what do you get if you add together all of the rating numbers for all of the parts that ultimately get accepted?

Your puzzle answer was `446517`.

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

--- Part Two ---

Even with your help, the sorting process still isn't fast enough.

One of the Elves comes up with a new plan: rather than sort parts individually through all of these workflows, maybe you can figure out in advance which combinations of ratings will be accepted or rejected.

Each of the four ratings (`x`, `m`, `a`, `s`) can have an integer value ranging from a minimum of `1` to a maximum of `4000`. Of all possible distinct combinations of ratings, your job is to figure out which ones will be accepted.

In the above example, there are `167409079868000` distinct combinations of ratings that will be accepted.

Consider only your list of workflows; the list of part ratings that the Elves wanted you to sort is no longer relevant. How many distinct combinations of ratings will be accepted by the Elves' workflows?

Answer: [Submit]

Although it hasn't changed, you can still get your puzzle input.

You can also [Share] this puzzle.