2023-02

bensmith

2023-12-02

Day 2: Cube Conundrum

Part 1

You're launched high into the atmosphere! The apex of your trajectory just barely reaches the surface of a large island floating in the sky. You gently land in a fluffy pile of leaves. It's quite cold, but you don't see much snow. An Elf runs over to greet you.

The Elf explains that you've arrived at Snow Island and apologizes for the lack of snow. He'll be happy to explain the situation, but it's a bit of a walk, so you have some time. They don't get many visitors up here; would you like to play a game in the meantime?

As you walk, the Elf shows you a small bag and some cubes which are either red, green, or blue. Each time you play this game, he will hide a secret number of cubes of each color in the bag, and your goal is to figure out information about the number of cubes.

To get information, once a bag has been loaded with cubes, the Elf will reach into the bag, grab a handful of random cubes, show them to you, and then put them back in the bag. He'll do this a few times per game.

You play several games and record the information from each game (your puzzle input). Each game is listed with its ID number (like the 11 in Game 11: . . .) followed by a semicolon-separated list of subsets of cubes that were revealed from the bag (like 3 red, 5 green, 4 blue).

For example, the record of a few games might look like this:

```
##                                                                          V1
## 1                  Game 1: 3 blue, 4 red; 1 red, 2 green, 6 blue; 2 green
## 2        Game 2: 1 blue, 2 green; 3 green, 4 blue, 1 red; 1 green, 1 blue
## 3 Game 3: 8 green, 6 blue, 20 red; 5 blue, 4 red, 13 green; 5 green, 1 red
## 4 Game 4: 1 green, 3 red, 6 blue; 3 green, 6 red; 3 green, 15 blue, 14 red
## 5                  Game 5: 6 red, 1 blue, 3 green; 2 blue, 1 red, 2 green
```

In game 1, three sets of cubes are revealed from the bag (and then put back again). The first set is 3 blue cubes and 4 red cubes; the second set is 1 red cube, 2 green cubes, and 6 blue cubes; the third set is only 2 green cubes.

The Elf would first like to know which games would have been possible if the bag contained only 12 red cubes, 13 green cubes, and 14 blue cubes?

In the example above, games 1, 2, and 5 would have been possible if the bag had been loaded with that configuration. However, game 3 would have been impossible because at one point the Elf showed you 20 red cubes at once; similarly, game 4 would also have been impossible because the Elf showed you 15 blue cubes at once. If you add up the IDs of the games that would have been possible, you get 8.

Determine which games would have been possible if the bag had been loaded with only 12 red cubes, 13 green cubes, and 14 blue cubes. What is the sum of the IDs of those games?

```
parse <- function(frame) {
  out <- frame %>%
  mutate(gameId = str_extract(V1, '^Game (\\d+):', group = 1) %>% as.integer()) %>%
  mutate(reds = str_extract_all(V1, '\\d+(?= red)')) %>%
  unnest(reds) %>% mutate(reds = as.integer(reds)) %>% group_by(gameId, V1) %>% summarize(reds = max(re
  mutate(greens = str_extract_all(V1, '\\d+(?= green)')) %>%
  unnest(greens) %>% mutate(greens = as.integer(greens)) %>% group_by(gameId, V1, reds) %>% summarize(g
  mutate(blues = str_extract_all(V1, '\\d+(?= blue)')) %>%
  unnest(blues) %>% mutate(blues = as.integer(blues)) %>% group_by(gameId, V1, reds, greens) %>% summar

  return(out)
}

exampleFrame <- parse(example)
```

```
## ‘summarise()‘ has grouped output by ’gameId’. You can override using the
## ‘.groups‘ argument.
## ‘summarise()‘ has grouped output by ’gameId’, ’V1’. You can override using the
## ‘.groups‘ argument.
## ‘summarise()‘ has grouped output by ’gameId’, ’V1’, ’reds’. You can override
## using the ‘.groups‘ argument.
```

```
exampleFrame
```

```
## # A tibble: 5 x 5
## # Groups:   gameId, V1, reds [5]
##   gameId V1                                                  reds greens blues
##    <int> <chr>                                              <int>  <int> <int>
## 1      1 Game 1: 3 blue, 4 red; 1 red, 2 green, 6 blue; 2 gr~     4      2     6
## 2      2 Game 2: 1 blue, 2 green; 3 green, 4 blue, 1 red; 1 ~     1      3     4
## 3      3 Game 3: 8 green, 6 blue, 20 red; 5 blue, 4 red, 13 ~    20     13     6
## 4      4 Game 4: 1 green, 3 red, 6 blue; 3 green, 6 red; 3 g~    14      3    15
## 5      5 Game 5: 6 red, 1 blue, 3 green; 2 blue, 1 red, 2 gr~     6      3     2
```

```
examplePossible <- exampleFrame %>%
  filter(reds <= 12 && greens <= 13 && blues <= 14)

examplePossible
```

```
## # A tibble: 3 x 5
## # Groups:   gameId, V1, reds [3]
##   gameId V1                                                  reds greens blues
##    <int> <chr>                                              <int>  <int> <int>
## 1      1 Game 1: 3 blue, 4 red; 1 red, 2 green, 6 blue; 2 gr~     4      2     6
## 2      2 Game 2: 1 blue, 2 green; 3 green, 4 blue, 1 red; 1 ~     1      3     4
## 3      5 Game 5: 6 red, 1 blue, 3 green; 2 blue, 1 red, 2 gr~     6      3     2
```

```
part1Example <- sum(examplePossible$gameId)
part1Example
```

```
## [1] 8
```

```r
frame <- read.delim('2023-02-input.txt', sep = '\n', header=FALSE) %>% parse()
```

```
## `summarise()` has grouped output by 'gameId'. You can override using the
## `.groups` argument.
## `summarise()` has grouped output by 'gameId', 'V1'. You can override using the
## `.groups` argument.
## `summarise()` has grouped output by 'gameId', 'V1', 'reds'. You can override
## using the `.groups` argument.
```

```r
print(frame)
```

```
## # A tibble: 100 x 5
## # Groups:   gameId, V1, reds [100]
##    gameId V1                                                   reds greens blues
##     <int> <chr>                                               <int>  <int> <int>
##  1      1 Game 1: 2 red, 2 green; 6 red, 3 green; 2 red, 1 g~     6      3     2
##  2      2 Game 2: 2 green; 15 red, 5 blue; 2 green, 4 blue, ~    15      3     6
##  3      3 Game 3: 10 blue, 8 red, 5 green; 5 green, 3 red; 1~    12      6    10
##  4      4 Game 4: 2 green, 3 blue, 1 red; 17 green, 1 blue, ~     5     17     3
##  5      5 Game 5: 4 green, 13 red, 3 blue; 14 blue, 5 green,~    13      7    14
##  6      6 Game 6: 1 green, 4 red; 1 blue, 19 red, 5 green; 1~    19     19     2
##  7      7 Game 7: 1 blue, 3 red, 3 green; 4 green; 1 blue, 6~     6      5     1
##  8      8 Game 8: 1 green; 1 green, 16 red, 1 blue; 3 red, 1~    16      1     2
##  9      9 Game 9: 5 green, 1 red; 6 red, 4 blue, 1 green; 9 ~    11     12     6
## 10     10 Game 10: 12 green, 4 blue, 7 red; 6 blue, 2 red; 7~     7     12     6
## # i 90 more rows
```

```r
possible <- frame %>%
  filter(reds <= 12 && greens <= 13 && blues <= 14)

print(possible, n=100)
```

```
## # A tibble: 46 x 5
## # Groups:   gameId, V1, reds [46]
##    gameId V1                                                   reds greens blues
##     <int> <chr>                                               <int>  <int> <int>
##  1      1 Game 1: 2 red, 2 green; 6 red, 3 green; 2 red, 1 g~     6      3     2
##  2      3 Game 3: 10 blue, 8 red, 5 green; 5 green, 3 red; 1~    12      6    10
##  3      7 Game 7: 1 blue, 3 red, 3 green; 4 green; 1 blue, 6~     6      5     1
##  4      9 Game 9: 5 green, 1 red; 6 red, 4 blue, 1 green; 9 ~    11     12     6
##  5     10 Game 10: 12 green, 4 blue, 7 red; 6 blue, 2 red; 7~     7     12     6
##  6     12 Game 12: 8 red, 4 blue; 4 green, 1 red, 2 blue; 1 ~    12     11     4
##  7     13 Game 13: 5 green, 4 blue, 4 red; 4 red, 5 blue, 2 ~     4     10     5
##  8     18 Game 18: 1 blue, 9 green, 1 red; 8 green, 5 blue; ~     1      9     5
##  9     19 Game 19: 3 blue, 3 red; 1 blue, 2 red, 4 green; 4 ~     3      4     7
## 10     20 Game 20: 1 blue, 2 red, 6 green; 4 red, 10 green, ~    11     10     5
## 11     22 Game 22: 4 green, 13 blue; 13 blue, 10 green; 8 gr~     3     10    13
## 12     26 Game 26: 5 blue, 1 red; 9 blue, 1 green, 1 red; 1 ~     1      1     9
## 13     27 Game 27: 3 red, 4 blue, 2 green; 2 red, 3 blue; 1 ~     3      5     4
## 14     28 Game 28: 1 red, 3 green, 6 blue; 5 red, 2 blue, 3 ~     5      6     6
## 15     30 Game 30: 9 red, 11 blue, 6 green; 4 red, 6 green, ~     9      6    13
## 16     31 Game 31: 11 red, 3 green; 3 green, 2 red; 6 red, 4~    11      6     4
## 17     33 Game 33: 6 green, 2 blue, 2 red; 1 red, 3 green, 7~     4      6    10
## 18     34 Game 34: 4 blue, 6 red; 7 red, 5 green, 3 blue; 2 ~     7      5     5
## 19     35 Game 35: 8 green, 1 blue; 12 blue; 6 green, 13 blu~     1      8    13
## 20     37 Game 37: 3 green, 1 red, 2 blue; 5 blue, 2 green, ~     4      3     5
## 21     40 Game 40: 4 blue, 2 green, 8 red; 3 blue; 7 blue         8      2     7
## 22     41 Game 41: 3 blue, 10 green, 7 red; 1 blue; 8 green,~     7     10     4
## 23     42 Game 42: 6 green, 5 blue, 1 red; 5 blue, 2 red, 6 ~     4     10     5
## 24     45 Game 45: 12 blue, 6 green, 9 red; 5 red, 6 blue, 2~     9      7    12
## 25     46 Game 46: 7 blue, 1 red; 1 green, 2 blue, 2 red; 3 ~     3      1     7
## 26     49 Game 49: 5 green, 8 red, 2 blue; 2 blue, 2 red, 1 ~     8     12     3
## 27     50 Game 50: 7 green, 8 red; 2 blue, 6 green; 1 green,~     8      7     2
## 28     53 Game 53: 7 blue, 3 green; 1 red, 6 blue, 6 green; ~     1      6     7
## 29     55 Game 55: 1 blue, 5 green, 3 red; 3 green, 4 red; 6~     6      5     1
## 30     56 Game 56: 5 red, 1 blue, 2 green; 6 green; 4 red, 6~     5      6     2
## 31     57 Game 57: 6 green, 3 blue; 8 blue, 6 green, 5 red; ~     5      7    12
## 32     63 Game 63: 1 red, 1 blue; 1 green, 1 red; 1 green; 1~     2      1     1
## 33     64 Game 64: 5 red, 1 blue, 5 green; 10 red, 4 green, ~    10      5     2
## 34     65 Game 65: 1 red, 8 green, 10 blue; 11 blue, 5 green~     2      8    11
## 35     69 Game 69: 8 green, 12 red, 11 blue; 8 red, 2 blue, ~    12     12    11
## 36     70 Game 70: 2 blue, 8 green; 9 green; 2 red; 2 red, 5~     4      9     2
## 37     78 Game 78: 5 blue, 4 green, 4 red; 1 red, 9 green, 4~     4      9     5
## 38     79 Game 79: 5 blue, 6 red; 2 blue, 1 green, 9 red; 3 ~     9      3     5
## 39     83 Game 83: 6 blue, 2 red; 3 blue, 11 red, 1 green; 1~    12      1     6
## 40     85 Game 85: 11 green, 7 red; 7 green, 4 blue, 6 red; ~    10     11     7
## 41     89 Game 89: 1 green, 2 blue, 1 red; 10 blue; 4 blue; ~     1      2    14
## 42     91 Game 91: 1 green; 9 green, 2 red, 2 blue; 3 blue, ~     2     12     3
## 43     92 Game 92: 1 green, 4 blue, 10 red; 12 blue, 9 red, ~    10      3    12
## 44     93 Game 93: 7 blue, 5 red; 2 green, 1 blue, 6 red; 1 ~     6      2     7
## 45     99 Game 99: 3 blue, 8 green, 1 red; 4 green, 1 blue, ~     2      8     3
## 46    100 Game 100: 8 red, 2 blue, 1 green; 2 blue, 4 red, 2~     9      2     5
```

```r
part1 <- sum(possible$gameId)
part1
```

```
## [1] 2169
```

Part 2

The Elf says they've stopped producing snow because they aren't getting any water! He isn't sure why the water stopped; however, he can show you how to get to the water source to check it out for yourself. It's just up ahead!

As you continue your walk, the Elf poses a second question: in each game you played, what is the fewest number of cubes of each color that could have been in the bag to make the game possible?

Again consider the example games from earlier:

```
Game 1: 3 blue, 4 red; 1 red, 2 green, 6 blue; 2 green
Game 2: 1 blue, 2 green; 3 green, 4 blue, 1 red; 1 green, 1 blue
Game 3: 8 green, 6 blue, 20 red; 5 blue, 4 red, 13 green; 5 green, 1 red
Game 4: 1 green, 3 red, 6 blue; 3 green, 6 red; 3 green, 15 blue, 14 red
Game 5: 6 red, 1 blue, 3 green; 2 blue, 1 red, 2 green
```

In game 1, the game could have been played with as few as 4 red, 2 green, and 6 blue cubes. If any color had even one fewer cube, the game would have been impossible. Game 2 could have been played with a minimum of 1 red, 3 green, and 4 blue cubes. Game 3 must have been played with at least 20 red, 13 green, and 6 blue cubes. Game 4 required at least 14 red, 3 green, and 15 blue cubes. Game 5 needed no fewer than 6 red, 3 green, and 2 blue cubes in the bag. The power of a set of cubes is equal to the numbers of red, green, and blue cubes multiplied together. The power of the minimum set of cubes in game 1 is 48. In games 2-5 it was 12, 1560, 630, and 36, respectively. Adding up these five powers produces the sum 2286.

For each game, find the minimum set of cubes that must have been present. What is the sum of the power of these sets?

```
part2Example <- exampleFrame %>%
  mutate(power = reds * greens * blues)

part2Example
```

```
## # A tibble: 5 x 6
## # Groups:   gameId, V1, reds [5]
##   gameId V1                                              reds greens blues power
##    <int> <chr>                                          <int>  <int> <int> <int>
## 1      1 Game 1: 3 blue, 4 red; 1 red, 2 green, 6 blue~     4      2     6    48
## 2      2 Game 2: 1 blue, 2 green; 3 green, 4 blue, 1 r~     1      3     4    12
## 3      3 Game 3: 8 green, 6 blue, 20 red; 5 blue, 4 re~    20     13     6  1560
## 4      4 Game 4: 1 green, 3 red, 6 blue; 3 green, 6 re~    14      3    15   630
## 5      5 Game 5: 6 red, 1 blue, 3 green; 2 blue, 1 red~     6      3     2    36
```

```
sum(part2Example$power)
```

```
## [1] 2286
```

```
part2 <- frame %>%
  mutate(power = reds * greens * blues)

part2
```

```
## # A tibble: 100 x 6
## # Groups:   gameId, V1, reds [100]
##    gameId V1                                             reds greens blues power
##     <int> <chr>                                         <int>  <int> <int> <int>
##  1      1 Game 1: 2 red, 2 green; 6 red, 3 green; 2 re~     6      3     2    36
##  2      2 Game 2: 2 green; 15 red, 5 blue; 2 green, 4 ~    15      3     6   270
##  3      3 Game 3: 10 blue, 8 red, 5 green; 5 green, 3 ~    12      6    10   720
##  4      4 Game 4: 2 green, 3 blue, 1 red; 17 green, 1 ~     5     17     3   255
##  5      5 Game 5: 4 green, 13 red, 3 blue; 14 blue, 5 ~    13      7    14  1274
##  6      6 Game 6: 1 green, 4 red; 1 blue, 19 red, 5 gr~    19     19     2   722
##  7      7 Game 7: 1 blue, 3 red, 3 green; 4 green; 1 b~     6      5     1    30
##  8      8 Game 8: 1 green; 1 green, 16 red, 1 blue; 3 ~    16      1     2    32
##  9      9 Game 9: 5 green, 1 red; 6 red, 4 blue, 1 gre~    11     12     6   792
## 10     10 Game 10: 12 green, 4 blue, 7 red; 6 blue, 2 ~     7     12     6   504
## # i 90 more rows
```

```r
sum(part2$power)
```

```
## [1] 60948
```