Advent of Code 2019: Some solutions

Katie LeVan

This is a narrative of my solutions to the Advent of Code 2019 Challenge.

Santa has become stranded at the edge of the Solar System while delivering presents to other planets! To accurately calculate his position in space, safely align his warp drive, and return to Earth in time to save Christmas, he needs you to bring him measurements from fifty stars.

Collect stars by solving puzzles. Two puzzles will be made available on each day in the Advent calendar; the second puzzle is unlocked when you complete the first. Each puzzle grants one star. Good luck!

— Day 1: The Tyranny of the Rocket Equation —

The Elves quickly load you into a spacecraft and prepare to launch.

At the first Go / No Go poll, every Elf is Go until the Fuel Counter-Upper. They haven't determined the amount of fuel required yet.

Fuel required to launch a given module is based on its mass. Specifically, to find the fuel required for a module, take its mass, divide by three, round down, and subtract 2.

For example:

- For a mass of 12, divide by 3 and round down to get 4, then subtract 2 to get 2.
- For a mass of 14, dividing by 3 and rounding down still yields 4, so the fuel required is also 2.
- For a mass of 1969, the fuel required is 654.
- For a mass of 100756, the fuel required is 33583.

The Fuel Counter-Upper needs to know the total fuel requirement. To find it, individually calculate the fuel needed for the mass of each module (your puzzle input), then add together all the fuel values.

What is the sum of the fuel requirements for all of the modules on your spacecraft?

First, I went ahead and input the puzzle data. Then I made a simple function that divided by 3, rounding down. Then I subtracted 2 from the result.

```r
# https://adventofcode.com/2019/day/1/input

## Input Data For Puzzle
day_1_puzzle = read.delim(paste(wd,'data/day-1-input.txt',sep='/'), header = F,
                          col.names = 'mass', encoding = 'UTF-8')

## Functions
FuelRequired <- function(mass){
  # Fuel required for a module, is its mass, divided by three,
  # rounded down, and subtract 2.
  fuel <- floor(mass/3)-2
  return(fuel)
}
```

Sapply allows this function to be applied to my vector of masses, returning a vector of results. These are summed using the sum function from base R.

```
### Part 1: What is the sum of the fuel requirements for all of the modules on your spacecraft?
fuel_required <- sapply(day_1_puzzle, FuelRequired)

sum(fuel_required)
```

Result: 3273471

— Part Two —

During the second Go / No Go poll, the Elf in charge of the Rocket Equation Double-Checker stops the launch sequence. Apparently, you forgot to include additional fuel for the fuel you just added.

Fuel itself requires fuel just like a module - take its mass, divide by three, round down, and subtract 2. However, that fuel also requires fuel, and that fuel requires fuel, and so on. Any mass that would require negative fuel should instead be treated as if it requires zero fuel; the remaining mass, if any, is instead handled by wishing really hard, which has no mass and is outside the scope of this calculation.

So, for each module mass, calculate its fuel and add it to the total. Then, treat the fuel amount you just calculated as the input mass and repeat the process, continuing until a fuel requirement is zero or negative. For example:

- A module of mass 14 requires 2 fuel. This fuel requires no further fuel (2 divided by 3 and rounded down is 0, which would call for a negative fuel), so the total fuel required is still just 2.
- At first, a module of mass 1969 requires 654 fuel. Then, this fuel requires 216 more fuel (654 / 3 - 2). 216 then requires 70 more fuel, which requires 21 fuel, which requires 5 fuel, which requires no further fuel. So, the total fuel required for a module of mass 1969 is 654 + 216 + 70 + 21 + 5 = 966.
- The fuel required by a module of mass 100756 and its fuel is: 33583 + 11192 + 3728 + 1240 + 411 + 135 + 43 + 12 + 2 = 50346.

What is the sum of the fuel requirements for all of the modules on your spacecraft when also taking into account the mass of the added fuel? (Calculate the fuel requirements for each module separately, then add them all up at the end.)

I modified the function to accomodate the change.

```
FuelRequired <- function(mass){
  # Fuel required for a module, is its mass, divided by three,
  # rounded down, and subtract 2.
  fuel = floor(mass/3)-2

  # Fuel requires its own fuel by the same equation
  extra_fuel = fuel
  while((floor(extra_fuel/3)-2)>0){
    extra_fuel = (floor(extra_fuel/3)-2)
    fuel = fuel + extra_fuel
  }
  return(fuel)
}

fuel_required <- sapply(day_1_puzzle$mass, FuelRequired)
sum(fuel_required)
```

Adding a while block adjusts the function to continue adding extra fuel while the result of any fuel addition is positive.

Result: 4907345