

- `using PlutoUI`

`parse_file` (generic function with 1 method)

- `function parse_file(io::IO)`
- `line = readline(io)`
- `return split(line, ",") .|> n -> parse{Int16}(n)`
- `end`

`minimal_fuel_to_align` (generic function with 1 method)

- `function minimal_fuel_to_align(initial_state, cost_of_fuel)`
- `(max_state,) = findmax(initial_state)`
-
- *# +1 due to one-off index*
- `computed_state = zeros{Int64,}(max_state+1)`
- `for pos in initial_state`
- *# 1 one-off index*
- `computed_state[pos+1] += 1`
- `end`
-
- `function get_fuel_to_align(pos)`
- `fuel_cost = 0`
- `for (index, total_at_pos) in enumerate(computed_state)`
- `actual_pos = index - 1`
- `fuel_cost += cost_of_fuel(abs(actual_pos - pos)) * total_at_pos`
- `end`
-
- `return fuel_cost`
- `end`
-
- `current_min = typemax{Int64}()`
- `min_pos = 0`
- `for pos in 0:max_state`
- `fuel_to_align = get_fuel_to_align(pos)`
- `if fuel_to_align < current_min`
- `current_min = fuel_to_align`
- `min_pos = pos`
- `end`
- `end`
-
- `return current_min, min_pos`
-
- `end`

Problem 1

(352254, 383)

0.001504 seconds (1 allocation: 15.625 KiB)

```

• with_terminal() do
•   open("./Day7/prob_input.txt") do io
•     initial_state = parse_file(io)
•     @time minimal_fuel_to_align(initial_state, x -> x)
•   end
• end

```

Problem 2

(99053143, 504)

0.692923 seconds (12.92 M allocations: 197.233 MiB, 4.43% gc time)

```

• with_terminal() do
•   open("./Day7/prob_input.txt") do io
•     initial_state = parse_file(io)
•     @time minimal_fuel_to_align(initial_state, x -> x == 0 ? 0 : Int(x/2 * (1+x))
•   )
•   end
• end

```

(99053143, 504)

```

• 99053143, 504

```

(1, 2)

```

• x(y) = y; a,b = x.([1,2]); a,b

```