

Advent of code

Programmering II

Axel Karlsson

VT22 February 15, 2022

Inledning

I denna uppgift har jag hjälpt tomtenissarna att hitta nycklarna till tomtens släde. Jag har implementerat två funktioner som på olika sätt mäter förändringar i en mängd data.

Dag 1

Den första dagen skulle en funktion implementeras som räknade antalet gånger djupet nycklarna var på ökade. Jag använde en ackumulatorvariabel som räknade antalet gånger ett djupet var större än det innan, och en rekursiv funktion som gick igenom den givna datan.

```
# Skapa ackumulatorvariabel
def depth1(data) do depth1(data, 0) end
# Basfall: Bara ett element kvar
def depth1([_], acc) do acc end
# Rekursivt steg
def depth1([h1, h2 | t], acc) do
  case h1 < h2 do
    true -> depth1([h2 | t], acc + 1)
    false -> depth1([h2 | t], acc)
  end
end
end
```

När jag applicerade denna funktion på `input()` blev resultatet 1564 ökningar.

Dag 2

Den andra dagen skulle samma mätning göras men med ett 3 `measurment sliding window`. Jag använde samma idé som för dag 1 men jag adderade tre mätningar i följd.

```

# Skapa ackumulatorvariabel
def depth2(data) do depth2(data, 0) end
# Basfall: Endast tre element kvar
def depth2([_, _, _], acc) do acc end
# Rekursivt steg
def depth2([h1, h2, h3, h4 | t], acc) do
  case (h1 + h2 + h3) < (h2 + h3 + h4) do
    true -> depth2([h2, h3, h4 | t], acc + 1)
    false -> depth2([h2, h3, h4 | t], acc)
  end
end
end

```

När jag applicerade denna funktion på `input()` blev resultatet 1611 ökningar.

Sammanfattning

Denna uppgift var inte särskilt svår jämfört med de tidigare uppgifterna och jag har inte så mycket att säga om den.