

On planet lisp the other day there was an entry for a lisp job at Streamtech in the Netherlands. In their job posting they had links to a number of fun problems. I decided to take a stab at one of them and here is what I came up with.

The Cat in the Hat

Background

(An homage to Theodore Seuss Geisel)

The Cat in the Hat is a nasty creature, But the striped hat he is wearing has a rather nifty feature.

With one flick of his wrist he pops his top off.

Do you know what's inside that Cat's hat? A bunch of small cats, each with its own striped hat.

Each little cat does the same as line three, All except the littlest ones, who just say "Why me?"

Because the littlest cats have to clean all the grime, And they're tired of doing it time after time!

The Problem

A clever cat walks into a messy room which he needs to clean. Instead of doing the work alone, it decides to have its helper cats do the work. It keeps its (smaller) helper cats inside its hat. Each helper cat also has helper cats in its own hat, and so on. Eventually, the cats reach a smallest size. These smallest cats have no additional cats in their hats. These unfortunate smallest cats have to do the cleaning.

The number of cats inside each (non-smallest) cat's hat is a constant, N . The height of these cats-in-a-hat is $\frac{1}{N+1}$ times the height of the cat whose hat they are in.

The smallest cats are of height one; these are the cats that get the work done.

All heights are positive integers.

Given the height of the initial cat and the number of worker cats (of height one), find the number of cats that are not doing any work (cats of height greater than one) and also determine the sum of all the cats' heights (the height of a stack of all cats standing one on top of another).

The Input

The input consists of a sequence of cat-in-hat specifications. Each specification is a single line consisting of two positive integers, separated by white space. The first integer is the height of the initial cat, and the second integer is the number of worker cats.

Sample Input

```
216 125
5764801 1679616
0 0
```

Sample Output

```
31 671
335923 30275911
```

Solution - Math

The key to understanding this problem is to understand that the relationship of the levels given in the problem description.

$$H_{wc} = \frac{1}{N+1} H_{wc+1}$$

This equation is just as true for the bottom most level of the worker cats as it is for any of the rest of it. Here we know that $H_{wc} = 1$ and so we know that $1 = \frac{1}{N+1} H_{wc+1}$ and it follows that

$$H_{wc+1} = N + 1$$

and we can also derive

$$H_{wc+2} = (N + 1)^2$$

$$H_{wc+n} = (N + 1)^n$$

by substituting it back into the equation as we go up the levels.

As we go down the list we can see that the number of cats increases by powers so at the top level we have one cat (N^0). On the second level we have N^1 or N cats, on the third level we have N^2 cats and so on up to N^{Level} for the number of levels that we have. We can see the same kind of relationship going from the bottom up. On the lowest level, the level with the worker cats we start with a size of 1 or $(N + 1)^0$ on the next to the lowest level we have $(N + 1)^1$ and going up to the top level $(N + 1)^{Level}$ as we described above.

We are given numbers for both of these equations for the number of levels we have, the height of the top cat

$$H_{tc} = (N + 1)^{Level}$$

and the number of worker cats at the bottom

$$N_{wc} = N^{Level}$$

The numbers we really need is the number of cats in each hat N and the number of levels $Level$. If we play with these formulas we can say :

$$\sqrt[Level]{H_{tc}} = (N + 1)$$

and

$$\sqrt[Level]{N_{wc}} = N$$

and that for some $Level$:

$$\begin{aligned} \sqrt[Level]{H_{tc}} - \sqrt[Level]{N_{wc}} &= (N + 1) - N \\ &= 1 \end{aligned}$$

because we do not have a n-root function in our target computer language, we will rewrite the above expression

$$\sqrt[Level]{H_{tc}} - \sqrt[Level]{N_{wc}} \implies H_{tc}^{\frac{1}{Level}} - N_{wc}^{\frac{1}{Level}}$$

Lets try it for the first set of inputs:

$$\begin{aligned}
 H_{tc} &= 216 \\
 N_{wc} &= 125 \\
 216^{\frac{1}{1}} - 125^{\frac{1}{1}} &= 91 \\
 216^{\frac{1}{2}} - 125^{\frac{1}{2}} &\sim 3.52 \\
 216^{\frac{1}{3}} - 125^{\frac{1}{3}} &= 1
 \end{aligned}$$

So we can say that there are 4 levels ($0 \Rightarrow 3$) and we know that $N = \sqrt[3]{125} = 5$ and the height of the second level cats is $N + 1 = 6$

and this allows us to calculate the actual output that we need to know.

First we solve for the total number of all of the non working cats:

$$\begin{aligned}
 N_{nwc} &= \sum_{x=1}^{Level} N^{Level-x} \\
 &= \sum_{x=1}^3 5^{3-x} \\
 &= 5^2 + 5^1 + 5^0 \\
 &= 25 + 5 + 1 \\
 &= 31
 \end{aligned}$$

And then we solve for the total height of all of the cats:

$$\begin{aligned}
 H_{ac} &= \sum_{x=0}^{Level} N^{Level-x} (N + 1)^{|x-Level|} \\
 &= \sum_{x=0}^3 5^{3-x} 6^{|x-3|} \\
 &= 5^3 6^0 + 5^2 6^1 + 5^1 6^2 + 5^0 6^3 \\
 &= 125 \cdot 1 + 25 \cdot 6 + 5 \cdot 36 + 1 \cdot 216 \\
 &= 125 + 150 + 180 + 216 \\
 &= 671
 \end{aligned}$$

The Code

The following code doesn't follow exactly the formulas above. What it does is recursively descends until we find our level, and then it performs the sums on the two items we are interested in as it works it's way back up. While it is not tail recursive, the number of levels should be serviceable.

```
(defun catwalk (height workers)
  (if (eq height 1) ; kitty must work alone
      (list 0 1)
      (let ((retlist (catwalk-sub height workers 0)))
          (subseq retlist 0 2 ))))

(defun catwalk-sub (height workers level )
  (let ((power (if (not (zerop level)) (/ 1.0 level) 0)))
      (cond
        ((eq 1 (ceiling (- (expt height power) (expt workers power ))))
          (list 0 workers (round (expt workers (/ 1.0 level))) level))
        (t (destructuring-bind (sumworkers sumheight n depth)
            (catwalk-sub height workers (+ 1 level))
          (progn
            (print (list sumworkers sumheight n depth level))
            (list (+ (expt n level) sumworkers)
              (+ (* (expt n level) (expt (+ n 1) (- depth level)) ) sumheight)
              n depth)))))))

(catwalk 216 125)

(catwalk 5764801 1679616)
```