

Name: _____

CS 440 — Recursion

Mattox Beckman

Hercules has a job to do. He has to slay the Hydra. The Hydra has nine heads. These are not just any heads; they are “level-9” heads. If one of them is cut off, eight level-8 heads grow to replace it. If you chop one of these, seven level-7 heads show up. This continues as you would imagine, until you get to a level-1 head. If you chop that one off, nothing else grows to take its place.

The question is this: how many head-choppings does Hercules have to perform to kill the Hydra?

There are closed-form solutions to this, but this is a lecture about recursion, so use recursion to solve this. Write a function that will take a representation of the Hydra and compute the number of choppings that has to happen to kill it. It should do this by converting the representation to one with one fewer heads.

It’s not required, but I do recommend you use tail recursion. The initial hydra head count will be represented by `[9,0,0,0,0,0,0,0,0]`. It shows nine heads of level nine, and no heads of the lower levels. You might find this easier if you write a function called `chop` that takes an input head-count and makes an output head-count.

E.g.,

```
chop [9,0,0,0,0,0,0,0,0]
```

yields

```
[8,8,0,0,0,0,0,0,0]
```

Here is an optional helper function to determine the level of the largest head on the hydra.

```
maxlevel hydra = aux hydra (length hydra)
  where aux _      0 = 0
        aux []     _ = 0
        aux (0:xs) i = aux xs (i-1)
        aux (x:xs) i = i
```