

# CS150 - Lab Prep 10

Due: Friday Nov. 18, at the beginning of class

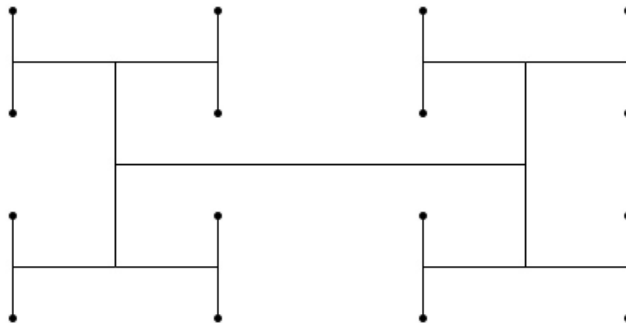
For this week's lab, we will be playing with recursion. For a majority of the lab, you will be using recursion to draw pictures using turtle graphics. To make sure that you can focus on the recursion, for the lab prep this week, we'll be working out some of the numerical details.

## 1. Recursive-H club

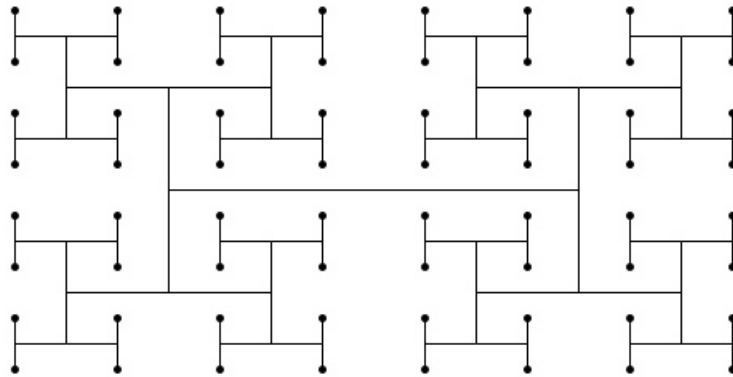
A recursive **H** is an **H** where the end of each vertical line of the **H** may have another recursive **H**. The recursive **H** is defined by a level. A level 1 recursive **H** is just an **H**:



A level 2 recursive **H** has another **H** at the end of each line:



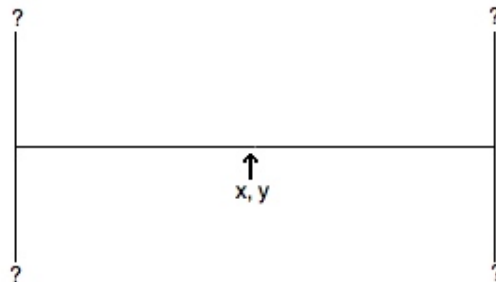
A level 3 recursive **H** has another **H** at the end of the each of these **H**s:



At the end, the smallest level of **H** always ends in a dot.

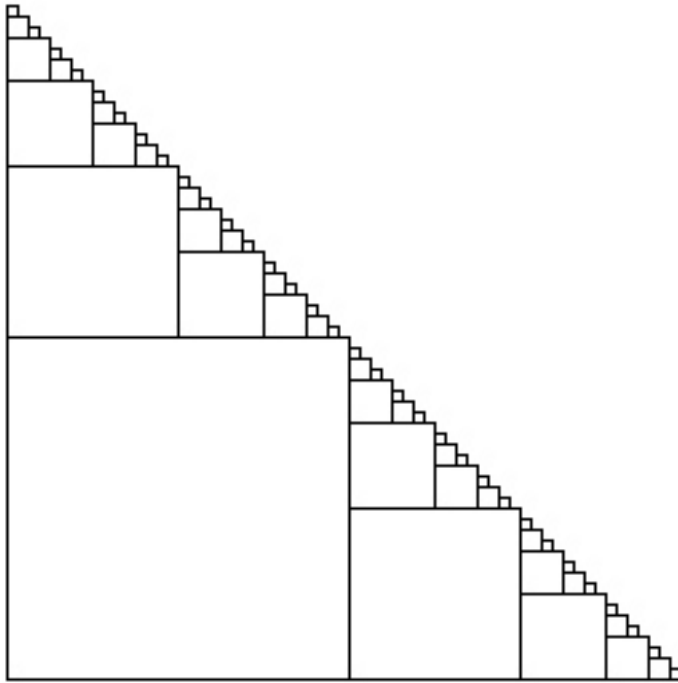
Each of these figures is made up of many **H**s. An **H** is defined by  $x, y$  coordinates and a length,  $l$ . The  $x, y$  coordinates are at the center of the **H**. The vertical bars are of length  $l$  and the horizontal bar of length  $2l$ .

**Given an **H** at  $x, y$  with length  $l$  what are the 4 coordinates of the ends of the vertical bars of that **H** (that is the upper left, lower left, upper right and lower right of the **H**)?**

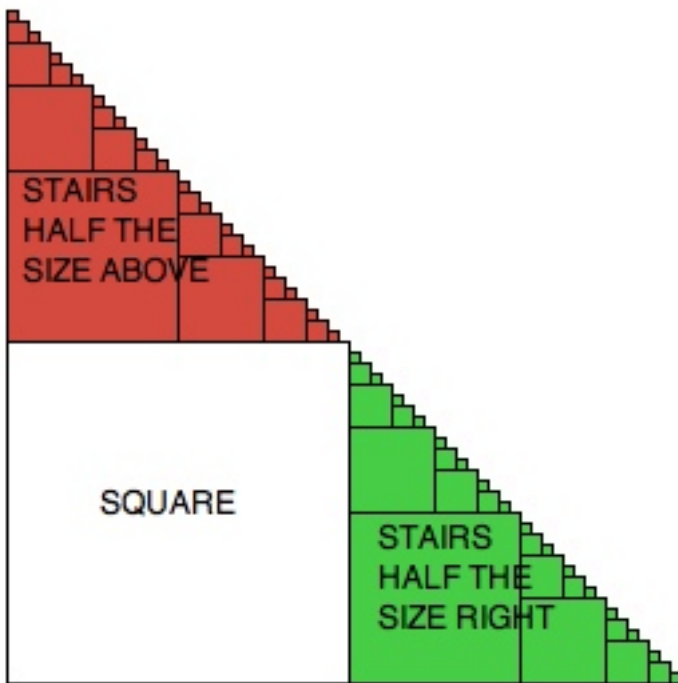


## 2. Stairs

We will also be creating a program that draws “stairs”:



To draw stairs, we first draw a large square in the lower left. We then draw a set of stairs above it and to the right that are half as large:



Stairs are specified by the  $x, y$  coordinates of the bottom left corner of the stairs and the length  $l$  of a size of the square in the lower left hand corner.

**If we drew stairs starting at  $x, y$  with length  $l$  what would be the coordinates of the recursive set of stairs above and to the right of the initial square?**

### 3. Color in turtle graphics

We've seen different fill colors in turtle graphics specified by a string (e.g. `fillcolor("yellow")` or `fillcolor("blue")`). In many situations, this is sufficient, however, in some cases we need a bit more granularity. Instead of passing a string to `fillcolor`, if you need more precision, you can pass three numbers between 0 and 1.0 specifying how the proportion of red, green and blue that make up the color (called rgb coloring). For example:

```
>>> fillcolor(0, 1, 0)
>>> begin_fill()
>>> circle(50)
>>> end_fill()
```

will draw a green circle. See what happens as you change these three numbers.

**What would be the rgb values if we wanted a light blue color?**