In[19]:=
```
(* Adam Beck *)
```

In[20]:=
```
(* Calculate the fractional dimension of the Sierpinski carpet and "sponge" *)

(* Sierpinski carpet *)
```

In[21]:=
```
(* consider a 2D square. As discussed in class, we need (1/epsilon)^2 objects
 to "fill" the square, where epsilon is the diameter. Let E refer to epsilon. *)
(* dimension = log[N{E,A}] / log[1/E] *)
(* log[N{E,A}] is calculated by taking the mass of the current iteration of the carpet,
and plugging E into that mass equation. *)

(* For the first iteration, the mass of the carpet is 8/9, the next is 8/9 squared,
then cubed, and so on. These are all multiplied by (1/E) squared. A
 For loop can be used to iterate through a few iterations of the carpet,
and converge on an answer. Let epsilon be raised to the 1,
2nd, 3rd, etc. power for every iteration *)

epsilon = 1 / 3;
mass = 8 / 9;

For[i = 1, i < 5, i++, (
   epsilonTemp = (epsilon ^i) ^2;
   massTemp = mass ^ i;
   massEquation = massTemp / epsilonTemp;
   Print[N[Log[massEquation] / Log[1 / (epsilon^i)]] ])
 ];
```

```
1.89279

1.89279

1.89279

1.89279
```

In[24]:=
```
(* Therefore the dimension(A) =
 limit as E approahces infinity of Log[N{E,A}] / log(1/E) = 1.89279 *)

(* Sierpinski sponge *)

(* This is relatively the same procedure. The "filling" is now (1/E)^3,
and the mass starts off at 26/27 for iteration 1, and cubes for every next iteration *)
epsilon = 1 / 3;
mass = 26 / 27;
```

In[26]:=

In[35]:=
```
For[i = 1, i < 4, i ++, (
   epsilonTemp = epsilon ^ (3^ (i - 1));
   massTemp = mass ^ (3 ^ (i - 1));
   massEquation = massTemp * ((1 / epsilonTemp) ^3);
   Print[N[Log[massEquation] / Log[1 / (epsilonTemp)]] ])
 ];
```

```
2.96565

2.96565

2.96565
```