Lecture notes: January 9, 2017

We covered the syllabus, including the use of GitHub for submitting programming assignments, and the course grading policies.

We also discussed the *phone dropping* problem:

Your goal is to figure out from how many stories high you can drop a phone from before it breaks. If you have one phone, the strategy is to start by dropping the phone from the 1st floor and work your way up floor by floor until the phone breaks. What is the optimal strategy (minimizes the worst case scenario of number of drops) if you have two phones? Assume you have a 100 story building.

Potential Approaches

Binary search (start at floor 50, then try floor 25 or 75, and continue, cutting it in half each time).
Issues: Usually, binary search is a great tool for these sorts of problems. The issue here is that you only have two phones, and once the first phone breaks you're stuck with just one phone (and hence have to use the one-phone solution of trying every floor). So if the first phone breaks when you try floor 50, you have to use the second phone to try every floor between 1 and 49 (worst case scenario: 50 drops total)

Intervals (Divide the building into equal intervals. Use the first phone to test the "top" of each interval until it breaks, use the second phone to test the floors in that interval).
Issues: This is a good general strategy.

A good guess might be to divide the building up into 10 even intervals. The first phone tests floors 10, 20, 30, etc., and if it breaks (say at floor 40), use the second phone to test floors 31, 32, etc. The worst case scenario here is that you need 19 drops (the first phone breaks on floor 100—10 drops—then you have to try 9 additional drops from floors 91 to 99).

In general, the worst case scenario is going to be (*number_of_floors*/*interval_size* + (*interval_size* - 1)).

Variable size intervals But we can do better. Note that if phone A breaks in the first interval, you only have to do 9 more drops with phone B for a total of 10 drops. If phone A breaks in the third interval, you still have to do 9 more drops with phone B for a total of 12 drops. Getting up to higher intervals means "spending" more drops with phone A before you use your 9 drops with phone B. If the final answer is in a higher interval, it will take you more drops to find it.

But what if we make higher intervals smaller? That way we "spend" drops getting to the kth interval, and then use fewer drops exploring that interval: If the final answer is in the kth interval, the number of drops needed to explore that interval is: k + size of kth interval.

So to minimize the worst case scenario, we want k + size of kth interval to be constant for all k.

That means we want the size of interval 1 to be 1 bigger than the size of interval 2, which should be 1 bigger than the size of interval 3, and so on.

Flipping things around, we want the *last* interval to only have 1 story in it, and the *second-to-last* interval to only have 2 stories in it, and so on.

All that is left is to figure out how many intervals we need to cover all 100 stories!

How many stories do we cover if we have *n* intervals? We can turn to some simple math:

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

(Later in class, we'll use a technique called *induction* to prove that this is true)

So if we want to cover 100 stories, we need:

$$\frac{n(n+1)}{2} = 100$$

Or *n* = 14 (technically, n = 13.65 or so, but we can't have intervals that are fractional stories).

So we can create 14 intervals, with the top interval being 1 story, the second being 2 stories, etc. and then guarantee that we will need no more than 14 drops to figure out how high we can drop a phone from before it breaks.