Two common types of algorithms

Las Vegas Algorithm
Always tells you the right answer
Takes varying amounts of time.

Monte Carlo Algorithm
Usually tells you the right answer. Sometimes the wrong one.

Focus on an element

Let's focus on one element of the array x_1.

The recursion will stop when every element is all alone in their own subarray.

Call an iteration "good for x_1" if the array containing x_1 in the next step is at most $\frac{3}{4}$ the size it was in the current step.

Pivot here: might leave x_1 in a big subarray (if x_1 is big)

Pivot here: both subarrays $\leq 3/4$ size. Must be good for x_1.

Pivot here: might leave x_1 in a big subarray (if x_1 is small)

How many levels?

How many levels do we need to go?

Once x_1 is in a size 1 subarray, it's done. How many iterations does it take?

If we only had good iterations, we'd need

$$\left(\frac{3}{4}\right)^k n \leq 1 \Rightarrow n \leq \left(\frac{4}{3}\right)^k \Rightarrow k \geq \log_{4/3} n.$$

I want (at the end of our process) to say with probability at least <blah> the running time is at most $O(n \log n)$.
What's the probability of getting a lot of good iterations…what's the tool we should use?

pollev.com/robbie

Practice with conditional expectations

Consider of the following process:

Flip a fair coin, if it's heads, pick up a 4-sided die; if it's tails, pick up a 6-sided die (both fair)

Roll that die independently 3 times. Let X_1, X_2, X_3 be the results of the three rolls.

What is $\mathbb{E}[X_2]$? $\mathbb{E}[X_2|X_1 = 5]$? $\mathbb{E}[X_2|X_3 = 1]$?