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.

3

For Simplicity

We'll talk about how quicksort is really done at the end. For now an easier-to-analyze version:

```
if(elements remaining > 1)
pick a pivot uniformly at random
split based on pivot
sortedLeft = QuickSort(left half)
sortedRight = QuickSort(right half)
return (sortedLeft pivot sortedRight)
```

7



10

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 $\left(\frac{3}{4}\right)^k n \le 1 \Rightarrow n \le \left(\frac{4}{3}\right)^k \Rightarrow k \ge \log_{4/3} n$. I want (at the end of our process) to say with probability at least <blab> 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