ALGORITHMS, FALL 2018, HOMEWORK 5

Due Thursday, October 11 at 11:59pm.
Worth 1% of the final grade.

1. In Quicksort, we recursively pick pivots to partition our data set. Given n elements, we choose to call a pivot *balanced* if each side of the partition ends up with at least a constant fraction $\frac{1}{c} \cdot n$ of the input, for a constant c that is decided in advance. Otherwise, a pivot is called *unbalanced*. In both cases, the partition takes dn time, where $d = O(1)$.
   As we run Quicksort, suppose that every balanced pivot is followed by u unbalanced pivots, where u is a constant.

   a) Show that in the above conditions, Quicksort will take $O(n \log n)$ time. Here, I don't care at all about the leading constant. Exaggerate as much as you like.

   b) If you haven't already done this in part (a), show roughly what the effect of u, d and c is on the upper bound for the runtime. In other words, where are they to be found, as hidden constants in the O-notation? For example, you might get something like $O(d^3 2^u n \log cn)$. (hopefully you won't actually conclude this!)
   In (b) you can still simplify your expression along the way, but try to let these three constants survive for as long as possible without losing an excessive amount of accuracy.

   Whatever you do, this isn't a rigorous analysis of Quicksort. It just gives some intuition for a specific scenario.