1. Consider a directed graph with each edge assigned a nonnegative weight $D$ that reflects the difficulty of passing over that edge (perhaps modeling an obstacle course). Define the difficulty of a directed path to be the maximum of the difficulties of its edges. Give an efficient algorithm that computes a path from $s$ (a designated vertex) to each of the other vertices, such that the path to any given vertex has minimum difficulty among all such paths.

2. Consider expanding tree-based union-find programs by including the added operation, print-set(x), that prints each of the members of the set containing x in some arbitrary order. Show how this can be implemented with just one extra pointer per node, so that its running time is $O$(number of elements printed out), and none of the other operations are degraded by more than a constant factor in terms of efficiency.

3. Suppose that an operation has negative amortized cost with respect to a particular choice of potential function $P$ that starts out at 0, and always remains non-negative. What conclusion can be drawn?

4. Do an amortized analysis for the cost of incrementing a base 3 counter, where actual cost of an increment operation is defined to be the number of positions that change by the operation. Give a potential function for which the amortized cost of increment is $3/2$.

Do exercises 5.10 and 5.22 (part a) in the text.