

## 1 Reading.

Chapter 8, Chapter 7.

## 2 Problems.

1. In class we discussed the union-find algorithm and argued that the amortized runtime of both union and find operations is near constant. Unfortunately, initialization of the union-find data structure requires  $\Theta(n)$  time. If we are only going to perform  $m$  operations and  $m$  is far fewer than  $n$ ; this cost of initialization is unacceptable.

Describe changes you would make to the data structure to enable initialization in constant time. Ensure that your changes do not affect the asymptotic runtime of either union or find. Describe any modifications necessary to *create*, *union*, and *find* operations.

2. Problem 8.1.

(For part (a), “arbitrarily” means  $\text{union}(i,j)$  makes the tree containing  $j$  a subtree of the root of the tree containing  $i$ .)

3. An sorting algorithm is *stable* if elements with equal value are left in the same relative order. Which of the sorting algorithms discussed in class are stable (insertion-sort, merge-sort, quick-sort, and heap-sort). For each one, give a brief explanation of why.
4. Problem 7.33.