

CSCE 750, Fall 2021, Assignment 3

Due September 28

September 20, 2021

This assignment covers material from the lectures on Chapters 6 and 7, in preparation for Quiz 3.

Not in the textbook: Dagwood tries to make a heap as a ternary tree instead of a binary one. Show how to do this. Analyze it. Is it a good idea?

Page 153: Exercises 6.1-1, 6.1-4, 6.1-5

Page 156: Exercises 6.2-1 (list the element comparisons and swaps, in order), 6.2-5, 6.2-6

Page 159: Exercise 6.3-1 (list the element comparisons and swaps, in order)

Page 160: Exercises 6.4-1 (optional, because it repeats some parts of 6.2-1 and 6.3-1.)

Not in textbook: Blondie has an array $A[1, \dots, 2n]$, in which n of the elements are '0' and the other n elements are '1'. She wants to find the index of one of the 0's, and she uses the following randomized algorithm:

```
FindIndexOfZero(A)
while true do
  i = random integer between 1 and 2n
  if A[i] = 0
    return i
  end if
end while
```

Find a tight asymptotic bound, in terms of n on the **expected** run time of this algorithm.

Page 180: Exercises 7.3-2 (Write a recurrence for the number of random numbers generated, then solve that recurrence via the substitution method.)

Page 184: Exercises 7.4-1, 7.4-2

Not in textbook: Consider this randomized version of MERGESORT:

```
RandomizedMergeSort(A, l, r)
  if l < r then
    m = random integer in the range { 1, ..., r-1 }
    RandomizedMergeSort(A, l, m)
    RandomizedMergeSort(A, m+1, r)
    Merge(A, l, m, r)
  end if
```

Recall that l and r are the lower and upper limits of the portion of the array to be sorted. The only change from standard MERGESORT is that m is selected randomly, rather than dividing the array into two equal parts. Write and solve, using any appropriate method, a recurrence for the worst-case expected run time of this algorithm. Is this algorithm an improvement over the standard MERGESORT?