

CSCE 750, Homework 3

This assignment covers material from the lectures on Chapters 6 and 7, in preparation for Quiz 3. NIT means “not in textbook.”

NIT1: 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?

Pages 163–164: Exercises 6.1-1, 6.1-4 [3rd ed.: Page 153: Exercises 6.1-1, 6.1-4]

NIT2: In an array that is in sorted order a min-heap? [3rd ed.: Exercise 6.1-5]

Pages 166–167: Exercises 6.2-1 (list the element comparisons and swaps, in order), 6.2-6, 6.2-7 [3rd ed.: Page 156: Exercises 6.2-1, 6.2-5, 6.2-6]

Page 170: Exercise 6.3-1 (list the element comparisons and swaps, in order) [3rd ed.: Page 159: Exercise 6.3-1]

Page 172: Exercises 6.4-1 (optional, because it repeats some parts of 6.2-1 and 6.3-1.), 6.4-3 [3rd ed.: Page 160: Exercises 6.4-1, 6.4-3]

NIT3: 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 193: Exercise 7.3-2 (Write a recurrence for the number of random numbers generated, then solve that recurrence via the substitution method.) [3rd ed: Page 180: Exercises 7.3-2]

Page198: Exercises 7.4-1, 7.4-2 [3rd ed.: Page 184: Exercises 7.4-1, 7.4-2]

NIT4: Consider this randomized version of MERGESORT:

```
RandomizedMergeSort(A, left, right)
  if left < right then
    middle = random integer in the range { left,..., right-1 }
    RandomizedMergeSort(A, left, middle)
    RandomizedMergeSort(A, middle+1, right)
    Merge(A, left, middle, right)
  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?