COS 226 Fall 2017 Quiz 2 35 pts.; 35 minutes; 4 questions; 5 pages. 2017-11-07 2:00 p.m.

Name: ____

1. (1 pt.)
   - **Read all material carefully.**
   - Budget your time: 35 minutes, 35 pts. $\Rightarrow$ 1 min./pt. avg.
   - You may refer to your books, papers, and notes during this test.
   - No computer or network access of any kind is allowed (or needed).
   - Write, and draw, carefully. Ambiguous or cryptic answers receive zero credit.
   - Use the conventions used in class and the textbook for notation, algorithmic options, etc.

   Write your name in the space provided above.

2. (10 pts.) Consider the following code, adapted from, Figure 21.28 of the textbook:

```
1       // Standard heapsort.
2       public static <T extends Comparable<? super T>>
3       void heapsort(T[] a)
4       {
5           for(int i = a.length/2 - 1; i >= 0; i--)
6               percDown(a, i, a.length);
7           for(int i = a.length - 1; i > 0; i--)
8           {
9               swapReferences(a, 0, i);
10              percDown(a, 0, i);
11          }
12      }
```

   Suppose we modify the first for loop by replacing `a.length/2` with `a.length` on line 5. (Everything else is unchanged, including code invoked by the above.) Answer the following, with brief justifications for your answers.

   (a) Does the modified version always give correct results?

   (b) What is the asymptotic running time of the modified heapsort, and how does it compare with that of the unmodified heapsort?

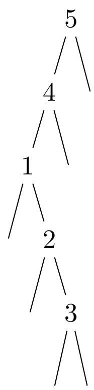
[additional space for answering the earlier question]

3. (10 pts.) Depict the action of *in-place heapsort* on the following array. Depict **both** the *state of the array*, and the *implicit binary heap it encodes* (in the usual graphical form), after each *deleteMin* operation.

   `54 19 61 91 53 15 47 29 48 60`

[additional space for answering the earlier question]

4. (14 pts.) Depict the transformations to the following *top-down splay tree* in response to the access pattern $1, 2, 3, 4, 5$. Depict all splay operations clearly, including the left and right trees, and highlight the tree after all operations for each insertion have completed.



[additional space for answering the earlier question]