Homework 1 (Ch. 2-4, 6-8)

Due June 28 at Midnight

Instructions Complete the following questions. Your submission should contain a single PDF with any pictures properly oriented. Additionally, you should modify the associated Java Gradle project and submit the project as a `.zip` file along with your PDF solutions.

Late homework will not be accepted. Hard to read homework will lose points.

1 Theory

1. CLRS 2.1-1
2. CLRS 2.1-4
3. Explain why $2n^2 \in O(n^2)$ is asymptotically tight but $2n \in O(n^2)$ is not. Why is $2n^2 \notin o(n^2)$? (Graphical explanations are OK)
4. Prove $\lg(n!) \in \Theta(n \lg n)$
5. Rank the following functions by their growth rate, where 1 grows the fastest, and two functions with the same growth rate have the same rank:
   - n
   - n^2
   - $\lg n$
   - $\lg \lg n$
   - $n \lg n$
   - $\sqrt{n}$
   - $\lg n^n$
   - $n!$
   - 2^n
   - n^n
6. CLRS 4.3-1
7. CLRS 4.3-2

8. CLRS 4.4-4
9. CLRS 4.5-1
10. CLRS 6.2-1
11. CLRS 6.4-3
12. CLRS 7.1-1 (only show the final partitioned array, not intermediate steps)
13. CLRS 7.1-3
14. CLRS 7.2-2
15. CLRS 8.3-1

2 Programming

Use the included Java project to implement the following functions (whose signatures have been provided for you).

- `insertionsort`
- `bubblesort` - I have implemented this for you
- `selectionsort`
- `mergesort`
- `quicksort`
- `heapsort`
- `countingsort`
- `bucketsort`

These should all be implemented in the file `src/main/java/homework1/Sort.java` and each should be implemented as a static method that modifies its input. The function signatures have been provided for you. Note that you may add as many helper functions as needed - the only requirement is that you do not modify the *signatures* of the required functions. For example, it may be helpful to create some helper functions when implementing quicksort and heapsort.

You can experiment with your sorts via `gradle run` and modifying the file `src/main/java/homework1/App.java`. To test your submission, run `gradle test`. Also note a helper method, `isSorted`, has been provided for you and is the method we will use to test whether an array is successfully sorted.

Two important notes:

1. **Code that does not build will receive a 0%.**
2. **Submissions that modify the given sort tests will receive a 0%.** If you would like to write additional tests, do so in a separate file or in `App.java`.