# Problem Set 3

Data Structures and Algorithms, Fall 2019

**Due: September 26, in class.**

## From CLRS

Exercise 6.1-7, 6.2-5, 6.3-3, 6.4-3, 6.4-5, 6.5-8. (For Exercise 6.5-8, also justify your `Heap-Delete` implementation is correct.) Problem 6-3.

## Additional Problem One

*[This is NOT a bonus problem, and you ARE required to solve it.]*

Professor F. Lake tells his class that it is asymptotically faster to square an $n$-bit integer than to multiply two $n$-bit integers. Should you believe him? Justify your answer.

## Additional Problem Two

*[This is NOT a bonus problem, and you ARE required to solve it.]*

The following cruel and unusual sorting algorithm was proposed by Gary Miller. Assume for this problem that the input size $n$ is always a power of 2.

```
Cruel($A[1 \cdots n]$)
if ($n > 1$) then
    Cruel($A[1 \cdots (n/2)]$).
    Cruel($A[(n/2+1) \cdots n]$).
    Unusal($A[1 \cdots n]$).
```

```
Unusal($A[1 \cdots n]$)
if ($n == 2$) then
    if ($A[1] > A[2]$) then
        Swap($A[1], A[2]$).
else
    for ($i = 1$ to $n/4$) do                                ▷ Swap 2nd and 3rd quarters.
        Swap($A[i + n/4], A[i + n/2]$).
    Unusal($A[1 \cdots (n/2)]$).
    Unusal($A[(n/2+1) \cdots n]$).
    Unusal($A[(n/4+1) \cdots (3n/4)]$).                      ▷ Recurse on middle half.
```

**(a).** Prove by induction that `Cruel` correctly sorts any input array.

**(b).** Prove that `Cruel` would *not* correctly sort if we remove the **for** loop from `Unusual`.

**(c).** Prove that `Cruel` would *not* correctly sort if we swap the last two lines of `Unusual`.

**(d).** What is the running time of `Unusual` and `Cruel`? Justify your answers.