Practice Assignment 4

Due: Tuesday, February 17 at 2PM to TEACH

To get credit, each student must submit their own solutions (which need not be typeset) to TEACH by the due date above – no late submissions are allowed. Note that while you must each submit solutions individually, you are free to work through the problem sets in groups. Solutions will be posted shortly after class and may be discussed in class. These will be not be graded on the basis of *completion* alone, not *correctness*.

1. Replace Thursday's lecture (which is cancelled) with the following activities:
   - Watch: `http://web.engr.oregonstate.edu/~glencora/cs325/videos/merge_sort.mp4`
   - Watch: `http://www.eecs.orst.edu/~glencora/cs325/videos/multiplication1_2.mp4`
   - Watch: `http://www.eecs.orst.edu/~glencora/cs325/videos/finishing_recursive.mp4`
2. Try to solve the following recurrence relations and give a Θ bound for each of them:
   (a) $T(n) = 2T(n/3) + 1$
   (b) $T(n) = 5T(n/4) + n$
   (c) $T(n) = 7T(n/7) + n$
   (d) $T(n) = 9T(n/3) + n^2$
   (e) $T(n) = 8T(n/2) + n^3$
   (f) $T(n) = 49T(n/25) + n^{3/2} \log n$
   (g) $T(n) = T(n-1) + 2$
   (h) $T(n) = T(n-1) + n^c$, where $c \geq 1$ is a constant
   (i) $T(n) = T(n-1) + c^n$, where $c > 1$ is some constant
   (j) $T(n) = 2T(n-1) + 1$
   (k) $T(n) = T(\sqrt{n}) + 1$
3. (a) Explain why the following algorithm sorts its input.

```
STOOGESORT(A[0...n-1])
    if n = 2 and A[0] > A[1]
        swap A[0] and A[1]
    else if n > 2
        k = ⌈2n/3⌉
        STOOGESORT(A[0...k-1])
        STOOGESORT(A[n-k...n-1])
        STOOGESORT(A[0...k-1])
```

   (b) Would STOOGESORT still sort correctly if we replaced $k = \lceil 2n/3 \rceil$ with $m = \lfloor 2n/3 \rfloor$? *(Hint: what happens when $n = 4$?)*
   (c) State a recurrence for the number of comparisons executed by STOOGESORT.
   (d) Solve the recurrence. Simplify your answer.