Fundamental Algorithms, Section 003
Homework 1, Fall 22.
Updated on Thursday, September 8.

Due date: Wednesday, September 14, 4:55pm on Gradescope

1.a. Consider the following code for the Insertion Sort algorithm.

```
1: Input: Array A[1 : n]
2: for j = 2 to n do
3:    i ← j − 1;
4:    while (i > 0 and A[i] > A[i + 1]) do
5:       swap(A[i], A[i + 1]);
6:       i ← i − 1
7:    end while
8: end for
```

Give a (reasonably tight) upper bound on the number of operations performed by this code on an input of n items. Assume the swap procedure performs 3 operations each time it is used. You should justify your answer with reasoning of the following form: each time line x is executed it performs y operations, for suitable x and y.

b. Specify a size n input on which the Insertion Sort algorithm performs a minimum number of operations. Modify your bound from part (a) to bound the number of operations performed on your example instance.

2. Consider the following pairs of functions. For each pair, state whether (i) $f = \Theta(g)$, (ii) $f = O(g)$ but $f \neq \Theta(g)$, (iii) $g = O(f)$ but $g \neq \Theta(f)$, or (iv) none of these.

a. $f(n) = n^2$, $g(n) = 4^{\log n}$.
b. $f(n) = n$, $g(n) = 2^{3 \log n}$.
c. $f(n) = 10n^2$, $g(n) = 10^{10}n$.
d. $f(n) = 3^n$, $g(n) = 3^{n \log n}$.
e. $f(n) = n^{2 \log n}$, $g(n) = (\log n)^n$.
f. $f(n) = n^4 + 3n^2 + 77$, $g(n) = n^4/1000$.
g. $f(n) = 5^n$, $g(n) = 4^n$.
h. $f(n) = \log \log n$, $g(n) = \log n$.
i. $f(n) = n$ when n is odd, $f(n) = n \log n$ when n is even; $g(n) = n$ when n is even, $g(n) = n^2$ when n is odd.
j. $f(n) = n$ when n is odd, $f(n) = n^2$ when n is even; $g(n) = n^2 + n$.

3.a. Let $f(n) = 5n^2 + 3n$ and $g(n) = 3n^3$. Show that $f(n) = o(g(n))$.

b. Show that $2^n = o(3^n)$.

4. Consider the problem of merging k sorted lists $L_1, L_2, \ldots, L_k$ of lengths $\ell_1 \leq \ell_2 \ldots \leq \ell_k$, respectively. Your procedure will repeatedly select two lists, merge them, and return the resulting merged list to the set of lists still to be merged, stopping when there is just one list left, i.e. after $k - 1$ merges.

Suppose that the merge of lists J and K takes $|J| + |K| + 1$ operations.

a. Suppose $k = 3$. What is the best order in which to perform the merges? Justify your answer.

b. Suppose $k = 4$. What is the best order in which to perform the merges? Justify your answer.

c. Challenge problem. This is not part of the homework and is not to be submitted. What is the best order for arbitrary k? Justify your answer.

The previous Problem 4, the Tower of Hanoi problem, is canceled and will appear on next week's homework instead.