

Put your *name* and *section number* on your solution and (if more than one sheet) staple. Turn in Section 2 of the NP-completeness homework separately.

1. Consider the following recurrence, defined for n a power of 5:

$$T(n) = \begin{cases} 19 & \text{if } n = 1 \\ 3T(n/5) + n - 4 & \text{otherwise} \end{cases}$$

Solve the recurrence exactly using the tree method (as done in class). Simplify as much as possible.

2. Consider a version of bubble sort that “bubbles” down the two largest elements (rather than just the one largest element). You can assume that the size of the list is even. To keep the code simple and in the spirit of standard bubble sort, only move elements by exchanging contiguous elements.
 - (a) Write the pseudo code.
 - (b) What is the worst-case number of comparisons? Justify.
 - (c) We can think of this as a recursive algorithm: Bubble the two largest elements to the bottom, and then recursively sort the rest of the list. Write the pseudo code for this recursive version.
 - (d) Based on your recursive algorithm, write a recurrence for the exact number of comparisons in the worst case.
 - (e) Solve the recurrence using the tree method. Note that the tree will be skinny.