

Problem Set 1

CS 331

Due Wednesday, January 26

General rules:

- For full credit, you must justify your work; if it is not obvious, a proof should be provided.
- Collaboration is encouraged, but you must write up the solutions on your own and acknowledge your collaborators at the top of your solutions.

1. Recursive time bounds: give a big-O bound for $T(n)$ given each of the following recursive formulas:

(a) $T(n) = 3T(n/4) + n \log n$

(b) $T(n) = 2T(n/2) + \sqrt{n}$

(c) $T(n) = 5T(n/4) + n$

(d) $T(n) = T(2n/3) + T(n/3) + n/6$.

with the base case $T(n) = O(1)$ for constant n .

2. [Exercise 0.2 of <http://jeffe.cs.illinois.edu/teaching/algorithms/>]

Careful readers might complain that our analysis of songs like “ n Bottles of Beer on the Wall” or “The n Days of Christmas” is overly simplistic, because larger numbers take longer to sing than shorter numbers. More generally, because there are only so many words of a given length, larger sets of words necessarily contain longer words. We can more accurately estimate singing time by counting the number of syllables sung, rather than the number of words.

- (a) How would you sing an arbitrary, very large integer n ? How many seconds does it take to sing, in big-Oh notation?
- (b) How long does it take to sing “ n Bottles of Beer on the Wall”?
- (c) How long does it take to sing “The n Days of Christmas”?

Express your answers in the form $O(f(n))$ for some function f .