Algorithms Lab 2: Recurrences

Review

Topics covered this week:

- Summations
- Recurrences
- Mergesort

Review the topics discussed in class this week and make sure you completed and thoroughly understand the pre-lab exercises. This is a great time to understand all the details and ask questions.

1 In lab exercises COLLABORATION LEVEL: 0

The in-lab problems are meant to be be solved during the lab and to generate discussion and sharing of ideas. Work with your team at your own pace and make sure to ask lots of questions.

Your answers will not be graded, however you need to work through these problems; please staple the answers to the in-class exercises to the assignment that you hand in.

1. Finish the recurrence exercises in the handout.
2. Give examples of recurrences that solve to logarithmic time.
3. Give examples of recurrences that solve to linear time.
4. Give examples of recurrences that solve to exponential time.

2 Homework problems COLLABORATION LEVEL : 1

The homework problem set is due next Friday. You are allowed and encouraged to collaborate while following the department's collaboration policy. Please refer to the class website for a description of the collaboration levels. List the people with whom you discussed the problems.

1. Find a tight bound for the solution of the following recurrences using iteration.

   (a) $T(n) = T(n/3) + 1$
   (b) $T(n) = T(n/3) + n$
   (c) $T(n) = T(\sqrt{n}) + 1$
   (d) $T(n) = T(n-1) + n$
   (e) $T(n) = 7T(n/2) + n^3$
   (f) $T(n) = 7T(n/2) + n^2$

2. (interview question) (a) Given an unsorted array and a number k. Find two elements in the array whose sum is k, or report if no such set exists. Analyze running time.

   (b) Generalize to 3-sum: Find if there exist 3 elements in the array whose sum is k, or report that no such subset exists. Analyze running time.

3. (interview question, 2014) Suppose you have an n-stories high building, and a bunch of eggs. An egg has a certain level l at which , if thrown from any level $\geq l$, it breaks. For example, an egg might have $l = 7$ meaning you can safely throw the egg down from levels 1 through 6, and it will not break; but if you through the egg from a level 7 or higher, it breaks.

   You are given a building and a bunch of eggs (all identical) and your goal is to find out the level l of the eggs. We can assume $n = 100$ (i.e. 100-level high building).

   (a) Describe an approach that only breaks one egg to find out l. How many throws does it do?
   (b) Describe an approach that minimizes the number of throws. How many eggs might it break?
   (c) Assume now you have two eggs. Describe an approach that minimizes the number of throws.

Notes on grading

You need to write each problem on a separate sheet of paper (do not forget to write your name). Each problem will be graded by a different TA.

Your assignment will be evaluated based not only on the final answer, but also on clarity, neatness and attention to details.

When you describe an algorithm, use high-level pseudocode. Focus on clarity, and don't forget to argue why the algorithm computes what it's supposed to compute (correctness), and to analyse its running time. You need to do this even if the problem does not specifically ask for it.

Generally speaking, it's not about getting the right answer, but about communicating it in a style that makes it easy to understand and convincing.