Algorithms[*] Lab 1: Analysis

Review

Topics covered this week:

- Case studies: bubblesort, insertion sort, selection sort
- Analysing best-case and worst-case running times
- Rate of growth
- Comparing the rate of growth of arbitrary functions
- Rate of growth of standard functions

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.

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.

1. Algorithm A uses $10n \lg n$ operations, while algorithm B uses n^2 operations. Determine the value n_0 such that A is better than B for $n \geq n_0$. The value does not have to be precise, for e.g. saying it's between 20 and 30 is good enough. Feel free to use a calculator, but you don't need to — plug in values to find the range of n_0. The goal of this problem is to reflect on $10n \lg n = O(n^2)$ and the fact that $10n \lg n > n^2$ for $n < n_0$.

2. Let $f(n) = \lg n$ and assume that we have an algorithm whose running time is $f(n)$ microseconds. Determine the largest size of a problem that can be solved by the algorithm in: (a) 1 second; (b) 1 hour; (c) 1 month; (d) 1 century.

   Same problem for $f(n) = n^{10}$ and $f(n) = 2^n$.

3. (CLRS 3-4.a) Prove or disprove: $f = O(g)$ implies that $g = O(f)$.

4. Prove or disprove: $f = O(g)$ implies that $g = \Omega(f)$.

[*]csci2200, Laura Toma, Bowdoin College

5. (interview question) You are presented with 9 marbles. All of the marbles look identical i.e. same shape, color, and dimensions(except for weight). However, 8 of the 9 marbles have exactly the same weight; the last marble is heavier. The only tool you have to measure weights is an old fashioned balance scale. You are only allowed to use the scale 2 times. How do you find the one marble that is not the same weight as the others?

6. (interview question) Close all notes, pick a language of your choice, and implement: (a) bubble sort; (b) insertion sort; (c) binary search. Include tester functions (generate random arrays etc).

   Do not open your notes! The goal is to be able to go on your own from the high level description of the algorithms (which you should know), to the low level details, which you need to figure out on the spot. To make it more fun, imagine an interviewer is looking at your screen.

Homework problems COLLABORATION LEVEL : 1

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 the order of growth of the following functions:

   (a) $n \lg \lg n + n \lg n + \sqrt{n} \lg^2 n$
   (b) $\sqrt{n} \lg n + n$
   (c) $n^2 + \sqrt{n} \lg^3 n$
   (d) $3^{\lg n} + n^2 + n \lg n$
   (e) $\sqrt{3}^{\lg n} + n^2 + n \lg n$
   (f) $2^n + 2^{2n}$
   (g) $2^{\lg n} + \lg n^2$
   (h) $(\lg n)^{\lg n} + n^3$

2. Describe a method for finding both the minimum and the maximum of n numbers with fewer than $3n/2$ comparisons.

3. Give an example of a positive function $f(n)$ such that $f(n)$ is neither $O(n)$ nor $\Omega(n)$.

4. Arrange the following functions in ascending order of growth rate. For each pair of consecutive functions, give a brief justification on why they are in this order. For e.g., if you ordered A, B, C, you need to justify that 1. $A = O(B)$; and 2. $B = O(C)$.

$$2^{\sqrt{\log n}}, 2^n, n^{4/3}, n(\log n)^3, n^{\log n}, 2^{2^n}, 2^{n^2}$$

5. Suppose each row of an $n \times n$ array A consists of 1's and 0's such that, in any row i of A, all the 1's come before any 0's. Assuming A is already in memory, describe a method running in $O(n)$ time (*not* $O(n^2)$ time) for finding the row of A that contains the most 1's.

6. Suppose you have algorithms with these five running times:

   (a) n^2
   (b) $100n^2$
   (c) n^3
   (d) $n \lg n$
   (e) 2^n

   How does the running time of these algorithms change when you double the input size?

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.