# Exercises

I. Due Monday, 2/2

p. 13, 1.2-2 (3 pts)

p. 27, 2.2-3 (4 pts,2 pts) Answer the first two questions: (a) How many elements need to be checked on the average (assuming the value that is being searched for is in the array, and is equally likely to be any one of the $n$ elements in the array)? (b) How many elements need to be checked in the worst case? Give exact answers (in terms of $n$) in each case.

p. 39, problem 2-4(a-c) (2 pts, 3 pts, 3 pts) In part (c), describe as precise a "relationship" as possible.

p. 50, 3.1-4 (4 pts) You only need to say yes or no in each case—no reasons are necessary.

p.58, 3-3(a) (15 pts) **Important: Please read these instructions carefully.** Omit all the functions involving $*$ (there are 4 of them), and number the remaining 26 from 1 through 26, going across the rows. For example, function number 4 is $(\lg n)!$, function number 15 is 1, function number 26 is $2^{2^{n+1}}$. Your answer should be the numbers 1 through 26 arranged in some order, so that if we considered the 26 corresponding functions in that order, the growth rates would be increasing. (For example, if the first three numbers in your answer were 26, 4, and 15, I would interpret this to mean that $2^{2^{n+1}} = O((\lg n)!)$ and $(\lg n)! = O(1)$. It is important that your answer be in precisely this form, because I will grade this problem by entering your sequence of numbers as input to a computer program. **Please make sure that your answer contains all 26 numbers exactly once. If your answer contains fewer than 26 numbers, I will enter some additional garbage values until there are 26.** Note: there is more than one correct answer, since there are a few pairs of functions in this list that are $O$ of each other, and in such a case, the two functions could appear in either order.

p. 59, 3-4 (8 pts) No proofs necessary, just say true or false. Make the assumption in each case that both $f$ and $g$ are increasing functions that approach $\infty$.

II. Due Friday, Feb. 13

General comment: simplify your answers as much as possible. For example, if the answer, is $O(n^2)$, don't write $O(n^{\log_2 4})$.

p. 57, exercise 3.2-4 (just yes or no) (3 pts each part)

p. 67, exercise 4.1-6 (4 pts)

p. 75, ex. 4.3-1 (2 pts each part), ex. 4.3-2 (4 pts)

p. 85, problem 4-1(c,d,e,f) (no justification necessary) (2 pts each)

p. 98, ex. 5.2-1 (1 pt for each question), 5.2-4 (4 pts)

III. Due Friday, March 6

p. 129, 6.1-1 (4 pts) Remember that a binary tree with 1 node has height 0. In both cases give exact answers in terms of $h$.

p.130, 6.1-4 (3 pts) Assume the question is asking for the array positions in which the smallest element might be. The positions in the array go from 1 through $n$.

p. 148, 7.1-2 (just the first question) (2 pts)

p. 149, 7.1-4 (3 pts) More precisely, indicate what change you could make to one line of the algorithm so that the effect would be to sort the array in nonincreasing order.

p. 153, 7.2-2 (3 pts) Give a $\Theta$-estimate. Assume the algorithm is the one shown on p. 146.

p. 161, 7-3(b) (4 pts)

p. 167, 8.1-1 (2 pts) The depth of a leaf means the number of levels below the root. (So the depth of the root node is 0.)

Assignment 4, due Monday, April 27

p. 179, 8-3(a) (10 pts) Write up your algorithm carefully, and give a clear, careful argument that the worst-case runtime is $O(n)$.

Assignment 5, due Friday, May 8

p. 259, 12.2-1 (4 pts), 12.2-4 (4 pts)
p. 530, 22.1-1 (4 pts), 22.1-7 (4 pts)
p. 548, 22.3-7 (3 pts), 22.3-8 (3 pts), 22.3-10 (3 pts)