Basic Algorithms – Midterm Exam

(1) Suppose the dictionary has been hashed to a table H such that probing costs constant time. (a) Show that the n-by-n word puzzle problem can be solved in $O(n^2)$ steps, by testing the presence of a word in H for each ordered quadruple (row, column, orientation, number of characters). (b) How do you refer to the runtime, as linear or quadratic? Hint: number of orientations is 8; maximum word size is some small constant.

(2) Design an algorithm to *Find* in a heap (and return positions in the table of) all nodes less than some value x. Your algorithm should run in $O(K)$, where K is the number of nodes whose positions are returned.

(3) Solve two of the following three recurrence equations for $W(n)$ at $n = 1+4k$, $T(n)$ or $S(n)$ at $n = 2^k$ (you are require to provide expressions for W, T or S in terms of n, not in terms of k).

$$\begin{array}{ll} (a) & W(i+4) = W(i) + i, \quad W(1) = 1. \\ (b) & T(2i) = 7 \cdot T(i) + i^2, \quad T(1) = 0. \\ (c) & S(2i) = r \cdot S(i) + 1/i, \quad S(1) = 0, \ 0 < r < 1. \end{array}$$

Hint to (b) and (c): Scale the variable i first and then treat the factors 7 or r. Make use of $\log_b(a) = 1/\log_a(b)$ when necessary.

(4) a) Prove by induction that a heap with n nodes has exactly $p = \lceil n/2 \rceil$ leaves.

b) Write a recursive algorithm for building a heap of size n in linear runtime.

(5) (Optional)

By inserting n numbers into a binary search tree and then performing an in-order traversal, we obtain a sequence of number so traversed. (a) Give a property of the sequence; (b) Show the runtime of the whole operation and justify your answer.