Department of Computer Science, National Tsing Hua University
Ph.D. Qualification Examination
Algorithms, Spring 2017

1. (10%) Growth of Functions
   a. (2%) Sort the following time complexity expressions in <u>ascending order</u>:

   $$O(n^2);\quad O(n);\quad O(n \log n);\quad O(\log n);\quad O(2^n)$$

   b. (3%) Determine the <u>tightest</u> big-O complexity of the recurrence:

   $$\mathrm{T}(n) = \mathrm{T}(n-1) + 1, \quad \mathrm{T}(0) = 0$$

   c. (3%) Determine the <u>tightest</u> big-O complexity of the recurrence:

   $$\mathrm{T}(n) = 2\mathrm{T}(n/2) + n^2, \quad \mathrm{T}(1)=0$$

   d. (2%) What is the <u>tightest</u> big-O complexity of finding the node with minimum key value in a min-heap of size N?

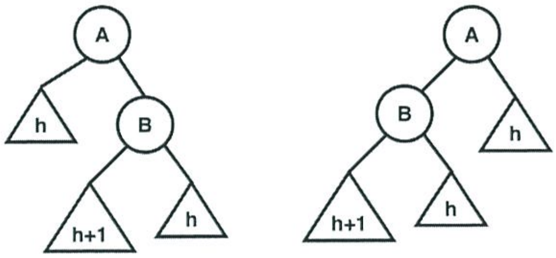
2. (10%) Sorting

   Given a list $L = \{12, 22, 18, 5, 8, 28, 6, 13\}$, write down the process of sorting L using:

   a. (5%) quick sort algorithm (the leftmost element is used as the pivot); and

   b. (5%) merge sort algorithm.

3. (16%) AVL Tree
   a. (8%) AVL tree is a self-balancing binary search tree (BST), which is capable of fixing the unbalanced sub-trees caused by BST insert/delete operations. Illustrate how AVL tree handles the following two unbalanced sub-trees.

   b. (8%) Draw the sequence of AVL trees by inserting the integer keys 9, 27, 50, 15, 2, 21, and 36 into an initially empty AVL tree.

4. (10%) Dynamic Programming

   Suppose a matrix-chain product has the following sequence of dimensions:

   $$\langle 4, 7, 10, 3, 50, 7, 8 \rangle.$$

   - That is, we want to multiply a sequence of matrices of dimensions 4×7, 7×10, 10×3, 3×50, 50×7, 7×8, respectively, and obtain a 4×8 matrix as the output.

   Use dynamic programming to find an optimal parenthesization of this product.

5. (10%) Greedy Algorithm
   a. (2%) Describe an $O(n)$-time algorithm that determines the smallest set of unit-length closed intervals to contain a sorted set of n 1-dimensional points $\{ x_1, x_2, \ldots, x_n \}$, where $x_1 < x_2 < \ldots < x_n$.
   b. (8%) Explain in details why your algorithm is correct.

6. (10%) Minimum Spanning Tree
   Given an edge-weighted connected graph $G = (V, E)$, where each edge has positive weight, the *bottleneck spanning tree* problem finds a spanning tree T of G such that the weight of T's maximum-weight edge is minimized. Describe an $O(|V|+|E|)$-time algorithm to solve the problem, and explain briefly why your algorithm is correct.

7. (10%) Polynomials and the FFT
   Explain how to use FFT to compute the polynomial multiplication for two degree-n polynomials in $O(n \log n)$ time.

8. (14%) Number-Theoretic Algorithms
   a. (5%) Explain how the RSA public-key cryptosystem works.
   b. (9%) Explain why the ability of factorizing the product of two large prime numbers efficiently can break the RSA public-key cryptosystem.

9. (10%) Approximation Algorithms
   Let G be a complete edge-weighted undirected graph whose edge weight satisfies the triangle inequality. Show that the following algorithm gives a 2-approximate solution of an optimal traveling-salesman tour.

1. Select a vertex as the root, and construct a minimum spanning tree *T* for *G*. 2. Traverse the tree *T* in preorder, and output the visited vertices as the solution of traveling-salesman tour.