IF184401 Design & Analysis of Algorithms (D)

Final Exam

Starting date:	3 April 2020
Deadline:	10 April 2020, 23:59 WIB. **Penalty: 0.15 points for each minute of tardiness.**
Exam type:	Open
Send to:	
	MM Irfan Subakti <yifana@gmail.com> CC to Agung Dwi Wicaksono <agung.dwi.temp@gmail.com> with the subject: IF184401_DAA(D)_FIN_StudentID_Name
File type and format:	A full report of the answers of the questions; in PDF format. Put this report along with your declaration into 1 (one) .ZIP file.
Filename format:	IF184401_DAA(D)_FIN_StudentID_Name.ZIP

Instruction
Please answer these questions as in the following.

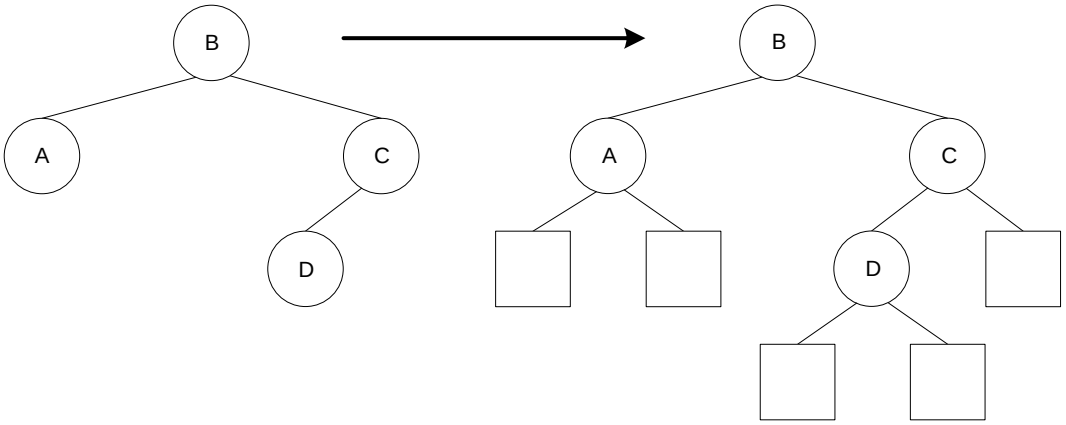
1. We started with easy questions. It's about True (T) or false (F) questions. **[15 points]**
   (1) Quick-sort has worse asymptotic complexity than merge-sort (T/F).
   (2) Binary search is $O(\log n)$ (T/F).
   (3) Linear-search in an unsorted array is $O(n)$, but linear-search in a sorted array is $O(\log n)$.
   (4) Linear-search in a sorted linked list is $O(n)$.
   (5) If you are only going to look up one value in an array, asymptotic complexity favours doing linear-search on the unsorted array over sorting the array and then doing a binary search (T/F).
   (6) If all arc weights are unique, the minimum spanning tree of a graph is unique (T/F).
   (7) Binary search in an array requires that the array is sorted (T/F).
   (8) Insertion into an ordered list can be done in $O(\log n)$ time (T/F).
   (9) A good hash function is one which tends to distribute elements uniformly throughout the hash table (T/F).
   (10) In practice, with a good hash function and non-pathological data, objects can be found in $O(1)$ time if the hash table is large enough (T/F).
   (11) If a piece of code has asymptotic complexity $O(g(n))$, then at least $g(n)$ operations will be executed whenever the code is run with parameter n.
   (12) Given good implementations for different algorithms for some process such as sorting, searching or finding a minimum spanning tree, you should always choose the algorithm with the better asymptotic complexity (T/F).
   (13) It is not possible for the depth-first and breadth-first traversal of a graph to visit nodes in the same sequence if the graph contains more than two nodes (T/F).
   (14) The maximum number of nodes in a binary tree of height H ($H = 0$ for leaf nodes) is $2^{H+1} - 1$ (T/F).
   (15) In a complete binary tree, only leaf nodes have no children (T/F).

2. Please write down a polynomial-time algorithm to find the longest monotonically increasing subsequence of a sequence of n numbers. Please explain in detail your assumption, the pseudo-code of this algorithm, and explain the time complexity of this algorithm. (Assume that each integer appears once in the input sequence of n numbers) **[20 points]**

3. After all, binary trees have been discussed a lot in our lecture. It's often used in search applications, where the tree is searched by starting at the root and then proceeding either to the left or the right child, depending on some condition. In some instances, the search stops at a node in the tree. However, in some cases, it attempts to cross a null link to a non-existent child. The search is said to "fall out" of the tree. The problem with falling

out of a tree is that you have no information about where this happened, and often this knowledge is useful information. Given any binary tree, an *extended binary tree* is one which is formed by replacing each missing child (a null pointer) with a special leaf node, namely *external node*. The remaining nodes are called *internal nodes*. An example is shown in the figure below, where the external nodes are shown as squares and the internal nodes are shown as circles. Note that if the original tree is empty, the extended tree consists of a single external node. Also, observe that each internal node has exactly two children and each external node has no children.

Let n denote the number of internal nodes in an extended binary tree. An extended binary tree with n internal nodes has $n + 1$ external node. Prove this statement by induction. **[20 points]**



4. We assumed that $G = (V, E)$ be a complete graph with the set of vertices $V = \{1, 2, \ldots, n\}$ and weights $w(i, j) = d_{ij}$ where d_{ij} are the distances from the input of *Traveling Salesman Problem* (TSP). Then we can do the following steps. (1) Select a vertex $r \in V$ to serve as a root for the future tree. (2) Build a minimum spanning tree T for G with the root r using a polynomial-time minimum spanning tree algorithm as a subroutine. (3) Construct the list L of vertices being a *Pre-order Walk* of T. (4) Return L as an approximate tour of G.

   Please prove the following statements.

   (1) The approximation algorithm for *Travelling Salesman* with triangle inequality is correct (i.e., produces a tour). **[5 points]**

   (2) The running time of the algorithm is polynomial. **[5 points]**

   (3) The algorithm has a ratio bound $\rho(n) = 2$. **[10 points]**

5. After we have finished answering question #4 above, then we can continue to answer this following question. If it's assumed that $P \neq NP$ then proves that there is no polynomial-time approximation algorithm with constant ratio bound ρ, for *Travelling Salesman Problem* (TSP). (**Clue**: *Hamiltonian Cycle*) **[25 points]**

Additional Instruction

Please do these steps as in the following.

A. To avoid plagiarism/cheating, every student needs to pledge and declare, then she/he must submit her/his **signed pledge and declaration** as in the following. Failed to do so will be resulted in getting 0 (zero) grade. Attach the **scanned**/**photo** of your *declaration* in your report.

"By the name of Allah (God) Almighty, herewith I pledge and truly declare that I have solved final exam by myself, didn't do any cheating by any means, didn't do any plagiarism, and didn't accept anybody's help by any means. I am going to accept all of the consequences by any means if it has proven that I have been done any cheating and/or plagiarism."

[Place, e.g., Surabaya], [date, e.g., 10 April 2020]

<Signed>

[Full name, e.g., Rahayu Sarasdewi]
[StudentID, e.g., 05111840000xxx]

B. Please create a full report, in PDF format, of the answers of the questions #1-5 above. Then put this PDF format report along with your declaration (see step A above) into 1 (one) .ZIP file. Please name this file: IF184401_DAA(D)_FIN_StudentID_Name.ZIP.

C. Have an amazing day, guys! Good luck! 😊