VANCOUVER ISLAND UNIVERSITY
CSCI 260 — FINAL EXAMINATION
10 December 2019, 9:00 — 12:00

Duration: 180 Minutes **Instructor:** H. Liu

TO BE ANSWERED IN BOOKLETS

Instructions

- Students must count the number of pages in this examination paper before beginning to write, and report any discrepancy immediately to the invigilator.
- This examination paper consists of 6 pages.
- This is a CLOSED BOOK examination. You are allowed to bring one piece of letter-sized and double-sided notes.
- Calculators, electronic devices and network connection access are NOT permitted.
- Remember to state any assumptions and show rough work.
- Note carefully the weight of each question, and answer appropriately.
- Attempt all questions. All questions relate to material covered in the lectures, labs and assignments.

Reference

The Master Theorem: Let $f(n)$ and $T(n)$ be defined as:

$$T(n) = \begin{cases} c & \text{if } n < d \\ aT(n/b) + f(n) & \text{if } n \geq d. \end{cases}$$

where $d \leq 1$ is an integer constant, $a > 0, c > 0$, and $b > 1$ are real constants and $f(n)$ is a function that is positive for $n \geq d$.

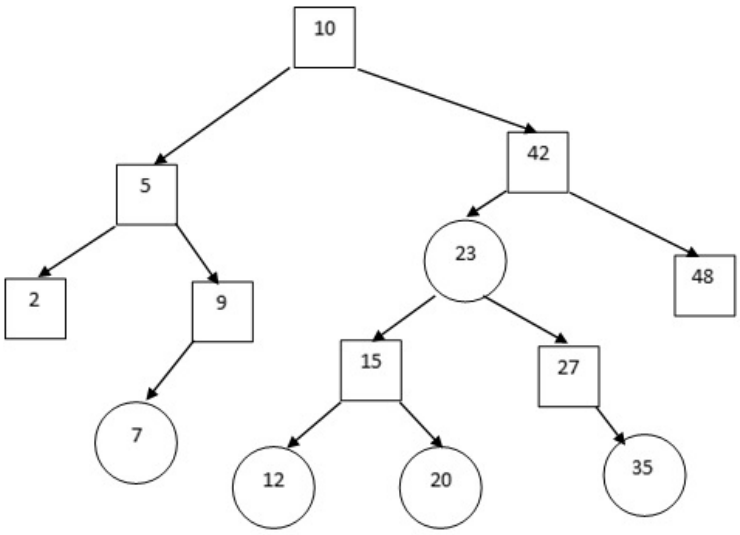
1. If there is a small constant $\varepsilon > 0$, such that $f(n)$ is $O(n^{log_b a - \varepsilon})$, then $T(n)$ is $\Theta(n^{log_b a})$.

2. If there is a constant $k \geq 0$, such that $f(n)$ is $\Theta(n^{log_b a} log^k n)$, then $T(n)$ is $\Theta(n^{log_b a} log^{k+1} n)$.

3. If there are small constants $\varepsilon > 0$ and $\delta < 1$, such that $f(n)$ is $\Omega(n^{log_b a + \varepsilon})$ and $af(n/b) \leq \delta f(n)$, then $T(n)$ is $\Theta(f(n))$.

1. (10 Marks) Suppose you are given an array A that contains N unique integer numbers in the range of [0..N] in sorted order (from smallest to largest). Note that there must be one number in the range [0..N] that is not in the array A.
   Describe an algorithm that runs in $\Theta(logN)$ time for finding the integer number that is not in array A.

2. (10 Marks) Given the following red-black tree (the square nodes are black nodes and the round ones are red nodes) that stores integer numbers:



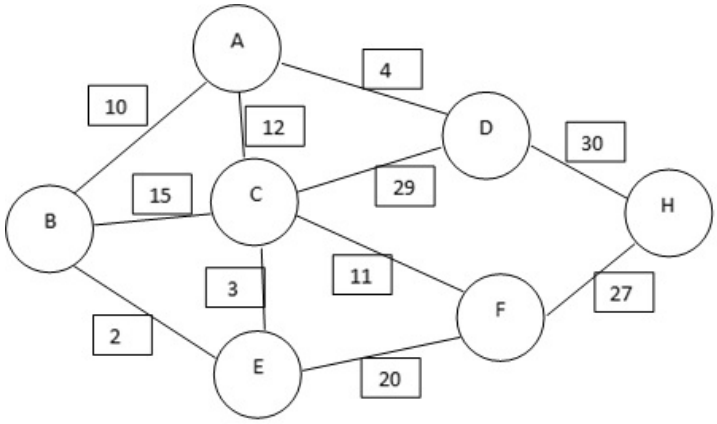
   (a) If we ignore the color property of the nodes in the above tree, is it also an AVL tree? Justify your answer.

   (b) Draw the (2,4)-tree that is equivalent to the above given red-black tree.

   (c) Draw the resulting red-black tree by inserting number 19 to the above given red-black tree. Clearly indicate whether/where the re-coloring and/or tri-node-restructuring operations is required.

3. (10 Marks) Write a C++ like algorithm that takes a string, an integer array whose size is 26 as its parameters and generates the frequency table (as the ones used in Huffman Encoding) for letters. This algorithm should be case-insensitive (i.e., treating upper-case letters the same as lower-case ones) and ignore all other characters that are not letters.

4. (10 Marks) Given the following undirected and weighted graph G:



   (a) Order the vertices as they are visited in the depth first traversal starting at vertex D. List all the **discovery** edges.

   (b) Draw the adjacency matrix for the above given graph G.

   (c) Show the execution steps of the Dijkstra's algorithm of single-source shortest path to all (reachable) vertices starting at vertex B.

5. (10 Marks) Suppose that you are given two arrays, *alphabet* and *code*. Each element of *alphabet* contains an alphabet character, and the same position element of *code* contains the Huffman code of that alphabet character. Each Huffman code is stored as a string of 0's and 1's. The two arrays have the same (positive) size and the size is known.
   Describe an algorithm that construct a Huffman tree based on these two arrays. You need to design a concrete data structure for each node of the Huffman tree first.

6. (10 Marks) Given the following strings and their corresponding hash codes, draw the resulting 13-cell hash table by inserting these strings in the given order to the hash table and using linear probing to resolve (possible) collisions.

string	**hash code**
brute	10
force	11
greedy	7
backtracking	10
dynamic	7
programming	8
abstract	0
methodology	9

7. (10 Marks) Given the following pattern string p:

   ```
   papaya paste
   ```

   (a) Draw the failure function of the above given pattern string p.

   (b) What is in general a failure function used in KMP string matching algorithm?

   (c) Why can using failure function improve the run time efficiency of the string matching algorithm?

8. (10 Marks) If you are asked to create a new abstract data type (for example, a Classroom ADT), how can you proceed to do it? what must be included in your description of the new ADT?

9. (5 Marks) If we have two algorithms that can both solve the same problem, one algorithm's run time efficiency is in $\Omega(N)$ and the other is in $O(N^2)$. Is there any way to judge which algorithm will give us better run time efficiency when the problem size N is really large? Justify your answer.

10. (10 Marks) Given the following recursive function `solve`:

```
void solve(int N)
{
    if (N <= 2)
        return;
    for(int i = 0; i < 3; i++) {
        solve(N/3);
    }
    f_combine(N);
}
```

Suppose the function `f_combine` is non-recursive and runs in $NlogN$ time.

(a) Give a recurrence equation $T(n)$ for the runtime of the function `solve`.

(b) Solve this recurrence equation using the Master Theorem. Explain your reasoning process briefly.

11. (5 Marks) The external sorting algorithm introduced in our lectures is modelled after the internal merge-sort algorithm. Is it possible to design an external sorting algorithm modelled after the internal quick-sort algorithm?
If your answer is yes, describe such an external sorting algorithm. If your answer is no, explain why such an external sorting algorithm is impossible.

====== END OF EXAM QUESTIONS ======