CSC B63 Winter 2023
Midterm Test
Duration — 90 minutes
Aids allowed: none

*Do **not** turn this page until you have received the signal to start.*

Please fill out the identification section above and read the instructions below.

Good Luck!

This exam is double-sided, and consists of **??** question. *When you receive the signal to start, please make sure that your copy is complete.*

- Please, make sure to **NOT write anything in the QR code areas**.
- Please, use a **black** or **blue pen** or a **thick in diameter pencil** to answer **all questions** in this booklet.
- If you use any space for rough work, indicate clearly what you want marked.

Question 1. Asymptotic Bounds [10 MARKS]

Part (a) [1 MARK]

Complete the following definition:

Definition 1. *Let $g \in \mathcal{F}$, where $\mathcal{F}$ is the set of all functions $f : \mathbb{N} \to \mathbb{R}^+$. Define $\mathcal{O}(g)$ to be the set of functions $f \in \mathcal{F}$ such that*

Part (b) [3 MARKS]

Give example functions $f, g \in \mathcal{F}$, such that $f \notin \mathcal{O}(g)$ and $g \notin \mathcal{O}(f)$. You do **not** need to provide proofs, but please provide an argument for why $f \notin \mathcal{O}(g)$ and $g \notin \mathcal{O}(f)$.

Question 1. (CONTINUED)

Part (c) [1 MARK]

Complete the following definition:

Definition 2. *Let $g \in \mathcal{F}$, where $\mathcal{F}$ is the set of all functions $f : \mathbb{N} \to \mathbb{R}^+$. Define $\Theta(g)$ to be the set of functions $f \in \mathcal{F}$ such that*

Part (d) [5 MARKS]

Using the above definition prove that $f_1 \in \Theta(f_2) \Rightarrow f_2 \in \Theta(f_1)$ for all functions $f_1, f_2 \in \mathcal{F}$.

Question 2. Balanced Trees [20 MARKS]

Part (a) [8 MARKS]

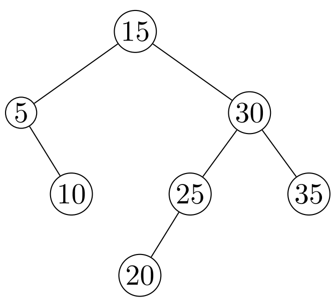
Provide pseudocode for an algorithm `isAVL` that takes (a root node of) a Binary Search Tree T and returns a tuple `(height, balanced)`, where (a) `height` is the number of nodes on the longest root-to-leaf path in T, and (b) `balanced` is $true$, if T is an AVL tree, and $false$, otherwise . Assume that each subtree S of T contains fields $S.left$ and $S.right$ for S's children, and that S is $null$ represents S is empty. **Your algorithm should visit each node in T *exactly once*.**

```
isAVL(T):
```

Question 2. (CONTINUED)

Part (b) [3 MARKS]

Show the result of inserting 21 and then 22 into the AVL tree below. You do **not** need to show intermediate steps, but in case your final answer is incorrect, intermediate work may earn you partial marks.

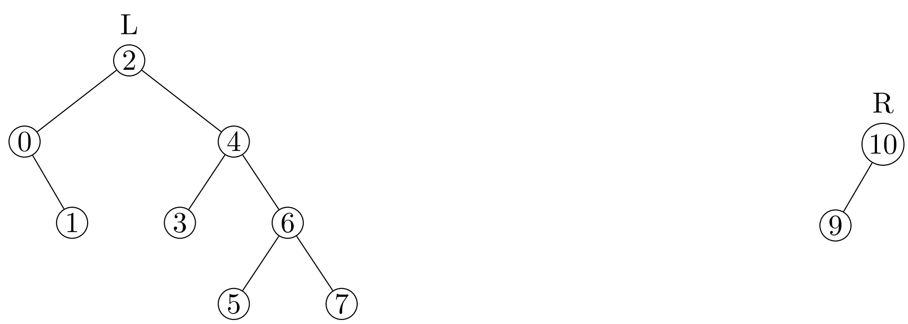


Part (c) [3 MARKS]

Give an example tree that is a weight-balanced tree, but not an AVL tree. Clearly demonstrate why your example is weight-balanced, and why it is not an AVL tree.

Question 2. (CONTINUED)

In class we saw that `join(L, k, R)` was a step in the algorithm for building unions of balanced trees. Consider the trees L and R below.



Part (d) [3 MARKS]

Using the algorithm for **AVL trees**, show the tree produced by `join(L, 8, R)`. You do **not** need to show intermediate steps, but in case your final answer is incorrect, intermediate work may earn you partial marks.

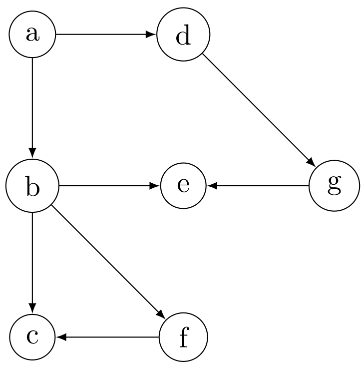
Question 2. (CONTINUED)

Part (e) [3 MARKS]

Using the algorithm for **weight-balanced trees**, show the tree produced by `join(L, 8, R)`. You do **not** need to show intermediate steps, but in case your final answer is incorrect, intermediate work may earn you partial marks.

Question 3. Graphs [12 MARKS]

Consider the following directed graph $G = (V, E)$ with its adjacency lists representation.



a	b, d
b	c, e, f
c	
d	g
e	
f	c
g	e

Part (a) [3 MARKS]

List vertices of V in order in which they are visited during the execution of Breadth-First Search started at vertex a. For each vertex, include the corresponding distance from a. Make sure to use the adjacency lists representation above.

vertex	a						
distance	0						

Part (b) [3 MARKS]

Show the *breadth-first tree* produced by the above execution of BFS.

Question 3. (CONTINUED)

Part (c) [3 MARKS]

List vertices of V in order in which they are visited during the execution of Depth-First Search started at vertex a. For each vertex, include its discovery time and finish time. Make sure to use the adjacency lists representation above.

vertex	a						
discovery	1						
finish	14						

Part (d) [3 MARKS]

Show the *depth-first tree* produced by the above execution of DFS.

Question 4. Heaps [10 MARKS]

Recall the array-based implementation of a Max-Heap we saw in class. In it the root node of a heap is stored at index 1.

Part (a) [1 MARK]

If a node is stored at index k, what is the index of its *left child*?

Part (b) [1 MARK]

If a node is stored at index k, what is the index of its *right child*?

Part (c) [1 MARK]

If a node is stored at index k, what is the index of its *parent*?

Part (d) [1 MARK]

What is the *minimum* number of nodes in a heap of height h (where h is the number of nodes on the longest root-to-leaf path)?

Part (e) [1 MARK]

What is the *maximum* number of nodes in a heap of height h (where h is the number of nodes on the longest root-to-leaf path)?

Question 4. (CONTINUED)

Part (f) [5 MARKS]

Prove that for any heap H, $height(H) \in \Theta(\log(size(H)))$ where $size(H)$ is the number of nodes in H.