

Student ID:

Student Name:

Course: Data Structures (CSE CS203A)

Assignment V: Tree

Due date: 2025.12.30 23:59:59

### **Important Notice – Use of AI Tools**

In this assignment, you must use at least one AI assistant (e.g. ChatGPT, Gemini, Claude, Grok, M365 Copilot) as a learning tool to help you:

- review definitions,
- compare tree variants, and
- organize your report.

You are not allowed to let the AI directly produce your final diagrams or final report content without your own understanding and rewriting.

You must log all AI prompts and services used (see “AI Usage Log” section below).

### **1. Goal of This Assignment**

In the lectures, we introduced the concept of the tree as a data structure, starting from the general tree and then moving to more specialized forms.

In this assignment, you will:

- a. Understand and clearly define:
  1. General tree
  2. Binary tree
  3. Complete binary tree
  4. Binary search tree (BST)
  5. AVL tree
  6. Red-Black tree
  7. Max heap
  8. Min heap
- b. Build a hierarchy and transformation path from the general tree to these variants, and explain how each variant adds more structure or constraints.
- c. Use a fixed list of integers to construct multiple tree variants and visualize them.
- d. Choose one real-world application for each tree type and explain why that data structure fits the application.
- e. Practice using AI tools as study companions and keep a simple Q&A log.

### **2. Given Data**

Use the following 20 integers as the input data for all your tree constructions:

37, 142, 5, 89, 63, 117, 24, 176, 58, 133, 92, 11, 151, 72, 39, 184, 7, 101, 54, 160

Student ID:

Student Name:

You will reuse this same sequence for every tree type (binary tree, complete binary tree, BST, AVL, Red-Black, max heap, min heap).

### 3. Deliverables

Please complete your work in the Student Worksheet Companion and upload it to the YZU Portal System.

Your report should include the following parts:

a. Definitions (Concept Review)

Provide clear, concise definitions for each of the following:

1. General tree
2. Binary tree
3. Complete binary tree
4. Binary search tree (BST)
5. AVL tree
6. Red-Black tree
7. Max heap
8. Min heap

You are encouraged to use AI tools to help you understand these concepts, but you must rewrite the definitions in your own words.

b. Hierarchy and Transformation of Tree Variants

Based on the definitions above, build a “tree family hierarchy” that shows how these structures are related. For example:

- General tree → Binary tree
- Binary tree → Complete binary tree / Binary search tree
- BST → AVL tree / Red-Black tree
- Binary tree → Max heap / Min heap

Tasks:

- Draw a diagram or flow chart that shows the transformation or specialization path: general tree → binary tree → complete binary tree → BST → AVL/Red-Black, etc.
- For each arrow (transformation), briefly explain:
  - What new constraint or property is added?
  - e.g., “Binary tree = tree with at most 2 children per node”,  
“BST = binary tree with left < root < right”,  
“AVL = BST with strict height-balance rule”, etc.

c. Tree Construction with the Given Integers

Using the given 20 integers, construct the following tree variants:

1. Binary tree
2. Complete binary tree
3. Binary search tree (BST)

Student ID:

Student Name:

4. AVL tree
5. Red-Black tree
6. Max heap
7. Min heap

Important Hint / Restriction:

- For these trees, you must use tree visualization tools (e.g., online visualizers or software) to build and display the tree.
- You may not ask AI tools to directly generate the final tree pictures for you.
- Instead:
  - Use AI only to help you understand algorithms,
  - Then apply those algorithms in a visualizer (or your own implementation).

What to submit for this part:

For each tree type:

- A snapshot (image) of the constructed tree.
- The URL / name of the visualization tool you used.
- A short note on how you inserted the integers (e.g., “insert in the given order as BST”, “build max heap using heapify”, etc.).

d. Application Example for Each Tree

For each of the following:

1. Binary tree
2. Complete binary tree
3. Binary search tree (BST)
4. AVL tree
5. Red-Black tree
6. Max heap
7. Min heap

Choose one application (real-world or system-level) and explain:

1. Application description
  - e.g., priority scheduling, dictionary lookup, memory allocation, database indexing, etc.
2. Why this tree structure fits
  - What property of this data structure makes it suitable?
  - Example:
    - Max heap → good for priority queue because the largest element is always at the root, so extracting max is efficient.
    - Red-Black tree → good for standard library maps/sets because it guarantees  $O(\log n)$  operations even under many insertions/deletions.

Student ID:

Student Name:

Your explanation should show that you understand the link between the data structure and its use case.

e. Report Layout and Organization

You are free to design the layout of your report, but it should:

- Be well-structured (use sections, headings, tables, and diagrams).
- Have a clear flow from:
  - definitions →
  - hierarchy/transformation →
  - constructed trees →
  - applications →
  - AI usage log.
- Be easy for another student to read and learn from.

Feel free to use AI to suggest a good outline, but you must decide and finalize the layout yourself.

f. AI Usage Log (Q&A Table)

Every time you use an AI copilot service for this assignment, record:

- Index (1, 2, 3, ...)
- Prompt (what you asked)
- Service (e.g., ChatGPT, Gemini, Copilot, ...)

Example log table:

Index	Prompt	Service
1	Assist me to have the definition of general tree, binary tree, complete binary tree, binary search tree, AVL tree, red-black tree, max heap and min heap for self-learning.	ChatGPT
2	Explain the difference between AVL tree and Red-Black tree in terms of balancing strategy and use cases.	Gemini
...	...	...

Place this table at the end of your report.

4. Evaluation (100 pts)

A possible breakdown (you can adjust if needed):

- Concept definitions (20 pts)
  - Correctness and clarity of all 8 tree type definitions.
- Hierarchy & transformation explanation (20 pts)
  - Clear diagram / explanation of how each tree variant evolves from the general tree.
  - Correct identification of constraints/invariants.
- Tree constructions & visualizations (25 pts)
  - Correct constructions for each tree type using the given integers.
  - Proper screenshots and tool URLs.

Student ID:

Student Name:

- Consistent insertion / heap-building strategy descriptions.
- d. Applications & explanations (20 pts)
  - One application per tree type.
  - Clear explanation linking data structure properties to the application.
- e. Report organization & AI usage log (15 pts)
  - Logical report structure and readability.
  - AI log completeness (all prompts listed with service names).
  - Thoughtful use of AI as a learning assistant, not as a copy-paste generator.

Student ID:

Student Name:

Course: Data Structures (CSE CS203A)

Assignment V: Tree

Student Worksheet Companion

Due date: 2025.12.30 23:59:59

## Academic Integrity and AI Usage Statement

In this assignment, you must use AI tools (such as ChatGPT, Gemini, Claude, Grok, M365 Copilot, etc.) as learning assistants, but you must also take full responsibility for understanding and organizing your own work.

### 1. Permitted Use of AI Tools

You may use AI to:

- Review or clarify definitions and concepts.
- Compare different tree data structures.
- Get suggestions for report layout or examples.
- Ask for explanations of algorithms (e.g., BST insertion, AVL rotation, heapify process).

You should read, think about, and rewrite the content in your own words.

### 2. Not Permitted

- Do not copy/paste AI-generated content directly as your final answer.
- Do not ask AI to draw the final diagrams or directly produce the final tree screenshots.
- Do not ask AI to complete the whole assignment report for you.

### 3. Your Responsibility

- You are responsible for understanding the definitions and algorithms.
- You are responsible for verifying whether AI answers are correct or not.
- You must produce your own original explanations and diagrams.

### 4. AI Usage Log

- You must record all AI queries related to this assignment.
- At the end of your report, include an AI Usage Log table with: Index, Prompt, AI service name.

By submitting this assignment, you acknowledge that you have used AI tools only as study aids, and that the final content of this assignment represents your own understanding and work.

## Section 1. Definitions of Tree Variants

Task: Write your own definitions for each tree type. You may use AI for learning, but rewrite in your own words.

### 1. General Tree

Definition:

Student ID:

Student Name:

2. Binary Tree

Definition:

3. Complete Binary Tree

Definition:

4. Binary Search Tree (BST)

Definition:

5. AVL Tree

Definition:

6. Red-Black Tree

Definition:

7. Max Heap

Definition:

8. Min Heap

Definition:

## Section 2. Tree Family Hierarchy and Transformations

Task: Show how these structures are related (general → specialized). Use a simple diagram and explanations of what constraints are added at each step.

### 2.1 Tree Family Diagram

You may draw this by hand and paste a photo, or use drawing tools.

Suggested chain example (you may extend or adjust):

General Tree → Binary Tree → Complete Binary Tree

Binary Tree → Binary Search Tree → AVL / Red-Black

Binary Tree → Max Heap / Min Heap

Your Diagram:

### 2.2 Explanation of Transformations

Fill in what new property or constraint is added at each step.

Student ID:

Student Name:

From	To	New property / constraint added
General Tree	Binary Tree	
Binary Tree	Complete Binary Tree	
Binary Tree	Binary Search Tree	
BST	AVL Tree	
BST	Red-Black Tree	
Binary Tree	Max Heap	
Binary Tree	Min Heap	

### Section 3. Tree Constructions Using Given Integers

Given integers (fixed for all parts):

37, 142, 5, 89, 63, 117, 24, 176, 58, 133, 92, 11, 151, 72, 39, 184, 7, 101, 54, 160

Task: For each tree type below, construct the tree using these integers, take a screenshot of the tree from your chosen tool, record the tool name/URL, and describe the insertion / heap-building procedure.

#### 3.1 Binary Tree

Tool name / URL:

Construction / insertion description:

Screenshot of Binary Tree (paste below):

#### 3.2 Complete Binary Tree

Tool name / URL:

Construction / insertion description:

Screenshot of Complete Binary Tree (paste below):

#### 3.3 Binary Search Tree (BST)

Tool name / URL:



Student ID:

Student Name:

Insertion rule (e.g., "insert in given order using BST rules"):

Screenshot of BST (paste below):

### 3.4 AVL Tree

Tool name / URL:

Insertion & balancing description:

Screenshot of AVL Tree (paste below):

### 3.5 Red-Black Tree

Tool name / URL:

Insertion & balancing description:

Screenshot of Red-Black Tree (paste below):

### 3.6 Max Heap

Tool name / URL:

Construction / heap-building description (e.g. heapify, insert-and-sift-up):

Screenshot of Max Heap (paste below):

### 3.7 Min Heap

Student ID:

Student Name:

Tool name / URL:

Construction / heap-building description:

Screenshot of Min Heap (paste below):

#### Section 4. Application Examples

Task: For each tree type, choose one application and explain why this tree is suitable.

Tree Type	Application Example (name / context)	Why this tree fits (properties that matter)
Binary Tree		
Complete Binary Tree		
Binary Search Tree		
AVL Tree		
Red-Black Tree		
Max Heap		
Min Heap		

#### Section 5. Reflection on Tree Family and Performance (Optional but recommended)

Among BST, AVL, and Red-Black trees, which one would you pick for:

Mostly search (few updates)? Why?

Frequent insertions and deletions? Why?

If you must store these 20 integers for static search only (no updates), which structure or representation would you prefer (sorted array + binary search, BST, AVL, etc.)? Why?

#### Section 6. AI Usage Log (Required)

Task: Record every time you ask an AI assistant about this assignment.

Index	Date / Time	AI Service (ChatGPT, Gemini, etc.)	Your Full Prompt / Question
1			
2			

Student ID:

Student Name:

3			
4			

You may extend this table as needed.