Student ID:1133346 Student Name:趙婉昀

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

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)

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.

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):

a. Concept definitions (20 pts)

- Correctness and clarity of all 8 tree type definitions.

b. Hierarchy & transformation explanation (20 pts)

- Clear diagram / explanation of how each tree variant evolves from the general tree.
- Correct identification of constraints/invariants.

c. Tree constructions & visualizations (25 pts)

- Correct constructions for each tree type using the given integers.
- Proper screenshots and tool URLs.

- 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:1133346 Student Name:趙婉昀

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:沒有限制的樹即每個節點子節點數量不固定，有階層結構 hierarchy，由節點、根等等組合成。

2. Binary Tree
   Definition:每個節點子節點最多只能為兩個的樹。

3. Complete Binary Tree
   Definition:全部節點都被兩個子節點填滿，除了最下面層的子節點都可以(可為 1)。

4. Binary Search Tree (BST)
   Definition:左節點小於父節點，右節點大於父節點。(每層都要遵守/二元)

5. AVL Tree
   Definition:左右子樹之間高度差不能大於 1。(有平衡/二元)

6. Red-Black Tree
   Definition:節點是紅色或黑色，紅色不能相連，根只能是黑色的。(有平衡/二元)

7. Max Heap
   Definition:最大值放最上面，父節點要大於子節點，二元。

8. Min Heap
   Definition:最小值放最上面，父節點要小於子節點，二元。

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

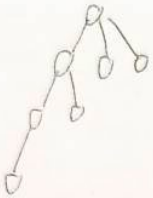
Your Diagram:

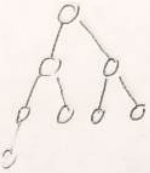
General Tree → Binary Tree → Complete Binary Tree

子節點數量不限

子節點最多有兩個

每一層都被填滿
除了最下面 leaf





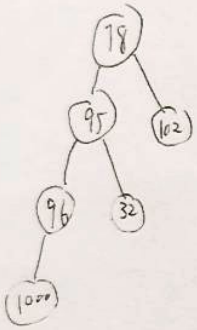
Binary Tree → Binary Search Tree → AVL / Red-black

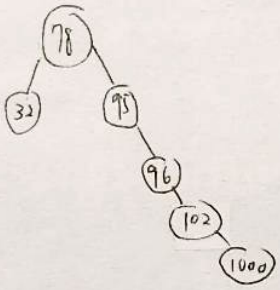
子節點為二
數值隨意放

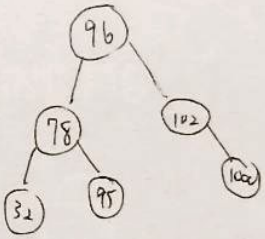
左子節點 < 父節點 < 右子節點

有平衡

AVL：左右子樹高度差最大為 1



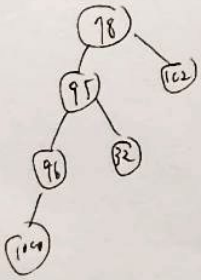




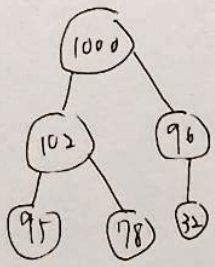
Binary Tree → Max Heap / Min Heap

子節點為二
數值隨意放

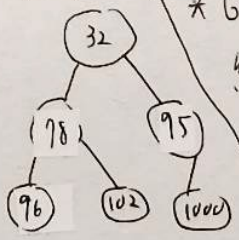
Max：父節點 > 子節點
Min：父節點 < 子節點



Max：



Min：



Red-black：將節點標為
紅色 or 黑色
root 必須為黑色
紅色不能相鄰
*GPT 補充：
經過的黑色節點
數量要一致

2.2 Explanation of Transformations

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

From	To	New property / constraint added
General Tree	Binary Tree	每個節點子節點只能有 0-2 個
Binary Tree	Complete Binary Tree	每一層都被填滿，除了最下面 leaf
Binary Tree	Binary Search Tree	左子節點<父節點<右子節點
BST	AVL Tree	左右子樹高度差最大為 1
BST	Red-Black Tree	節點標紅色或黑色、root 黑色、紅色不能相鄰，經過的黑色節點數量要一致
Binary Tree	Max Heap	父節點>=子節點 root 是最大值
Binary Tree	Min Heap	父節點<=子節點 root 是最小值

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

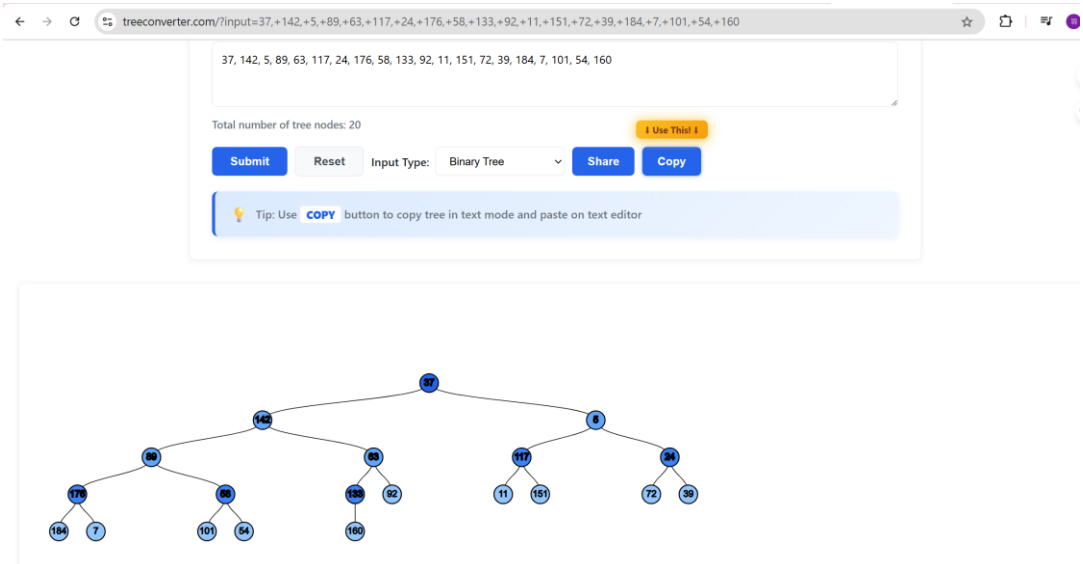
Tool name/URL:

https://treeconverter.com/?input=37,+142,+5,+89,+63,+117,+24,+176,+58,+133,+92,+11,+151,+72,+39,+184,+7,+101,+54,+160

Construction / insertion description:

照給的整數順序，從根一路往下，先填第一層左右子樹，再一路往第二層等等填上。

Screenshot of Binary Tree (paste below):



Student ID:1133346 Student Name:趙婉昀

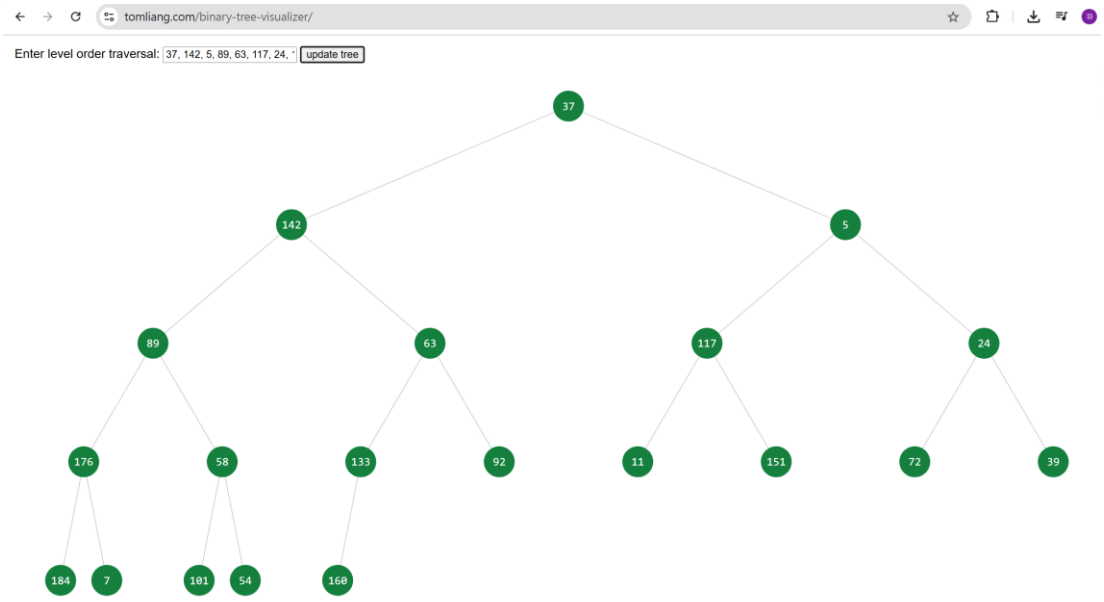
3.2 Complete Binary Tree

Tool name / URL: https://tomliang.com/binary-tree-visualizer/

Construction / insertion description:

只是一般生成二元樹的網站，沒找到特別標註 complete binary tree 的，但規則相同，會由上往下一層一層填，沒有填滿之前不會往下一層填，符合 complete binary tree 的定義，除了最下面一層均要填滿。

Screenshot of Complete Binary Tree (paste below):



3.3 Binary Search Tree (BST)

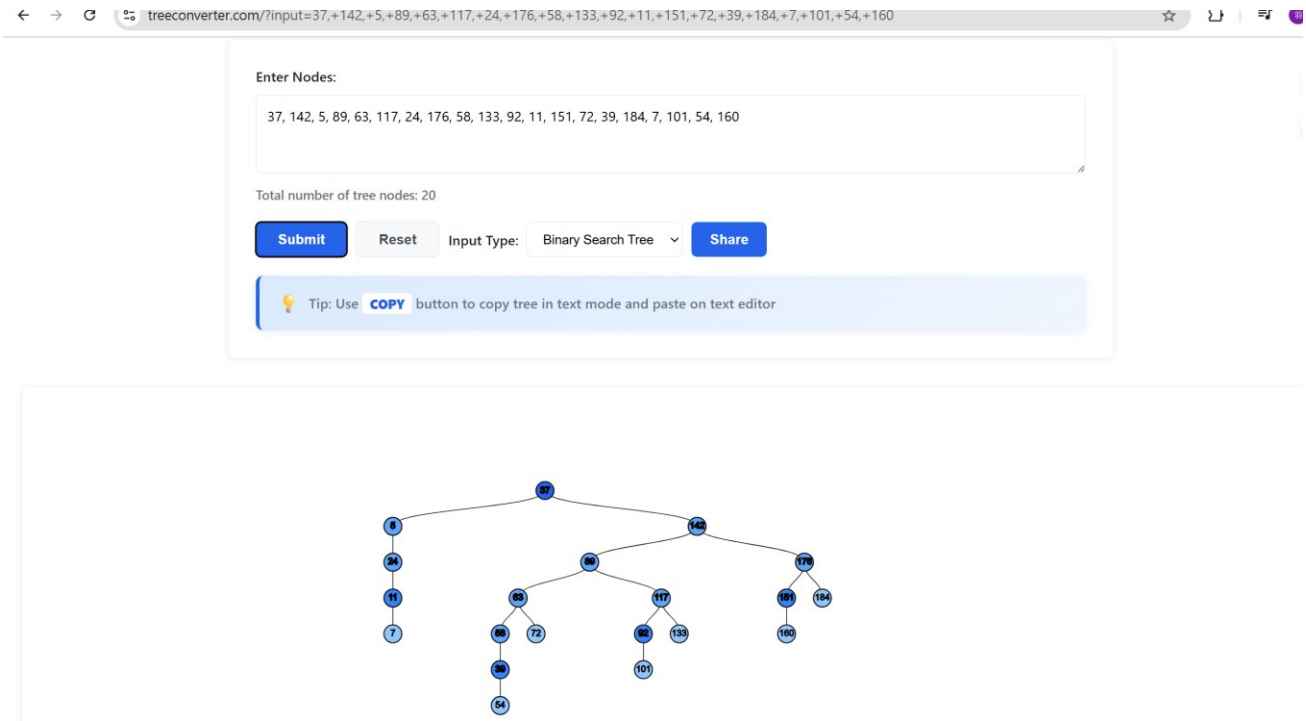
Tool name / URL:

https://treeconverter.com/?input=37,+142,+5,+89,+63,+117,+24,+176,+58,+133,+92,+11,+151,+72,+39,+184,+7,+101,+54,+160

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

照給的整數順序及 BST 規則，第一個當根，後面數值小於根往左子樹放，大於則放右子樹。

Screenshot of BST (paste below):



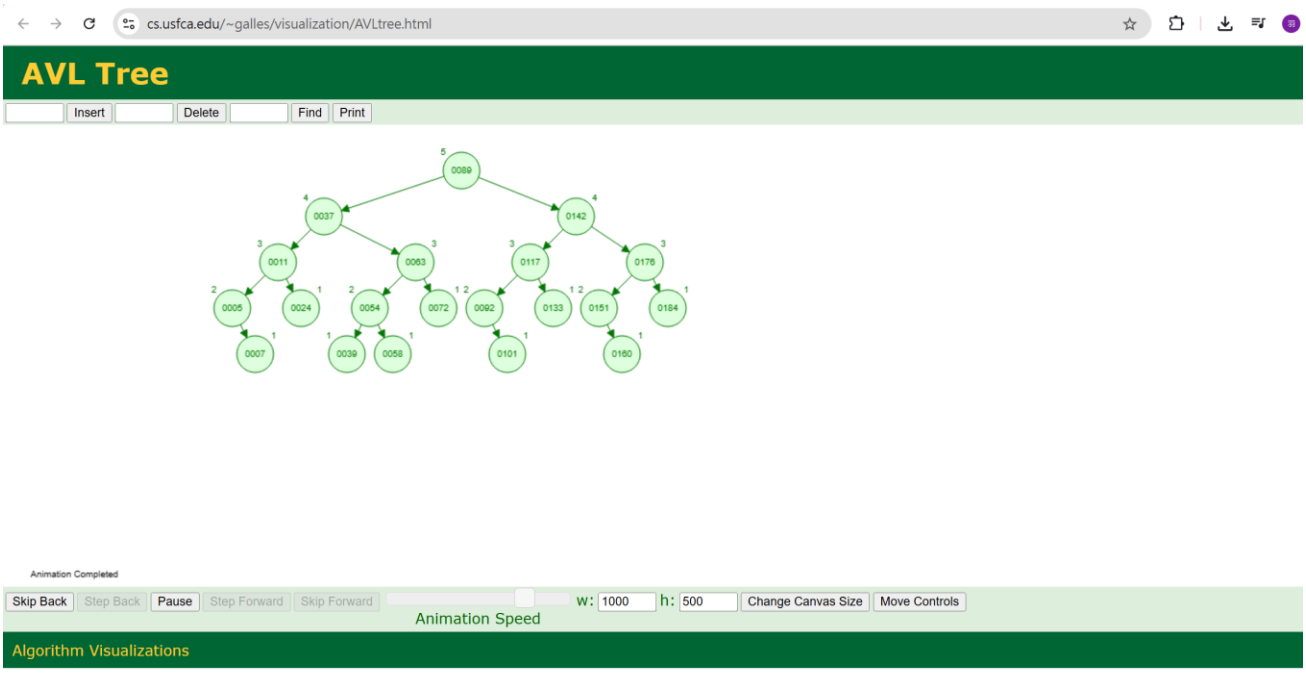
3.4 AVL Tree

Tool name / URL: https://www.cs.usfca.edu/%7Egalles/visualization/AVLtree.html

Insertion & balancing description:

先以 BST 規則:左<父<右，檢查是否符合左右差小於 1 並調整順序使其符合。(若不平衡，透過左旋或右旋進行調整。)

Screenshot of AVL Tree (paste below):



Student ID:1133346 Student Name:趙婉昀

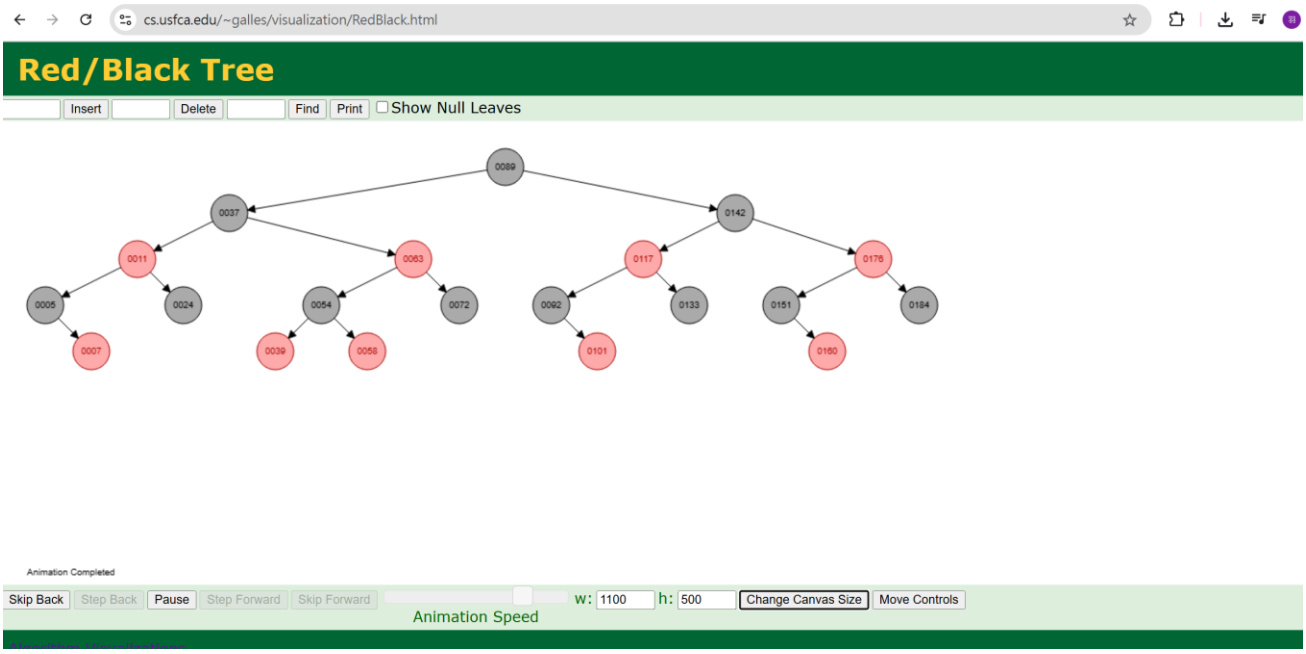
3.5 Red-Black Tree

Tool name / URL: https://www.cs.usfca.edu/~galles/visualization/RedBlack.html

Insertion & balancing description:

先以 BST 規則:左<父<右，檢查是否符合紅色不連續，由顏色跟旋轉調整順序。

Screenshot of Red-Black Tree (paste below):



3.6 Max Heap

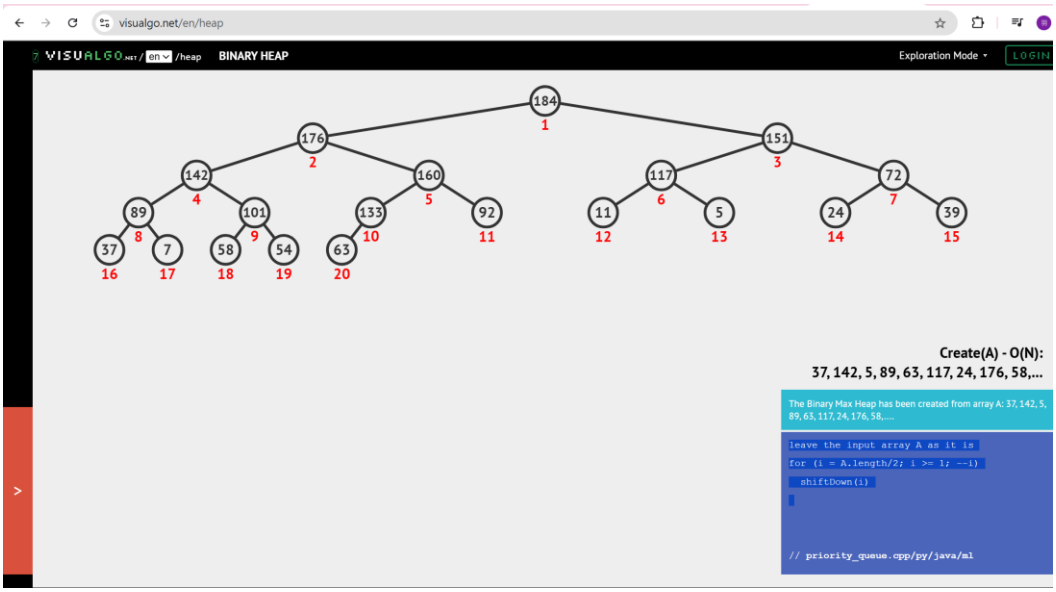
Tool name / URL:

https://visualgo.net/en/heap

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

先依序放入，再檢查是否都滿足父節點大於子節點，沒有的話就 sift-up 將兩者交換。

Screenshot of Max Heap (paste below):



3.7 Min Heap

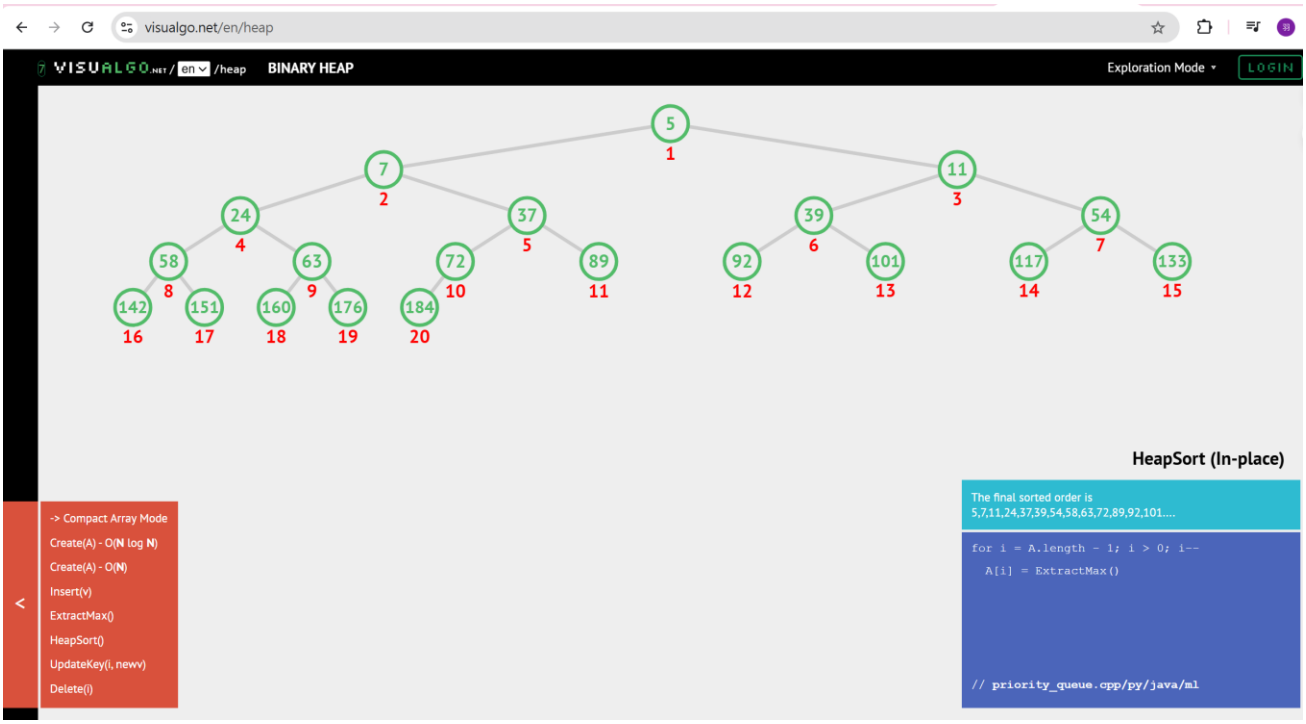
Tool name / URL:

https://visualgo.net/en/heap

Construction / heap-building description:

先依序放入，再檢查是否都滿足父節點小於子節點，沒有的話就 sift-down 將兩者交換

Screenshot of Min Heap (paste below):



Student ID:1133346 Student Name:趙婉昀

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	心理測驗/只回答是與否，最終引導到不同類型答案	因為是非題只有兩種回答，符合 binary tree，又因問題可能會直接在某個地方斷掉，如表單預計要女性則在該問題答 False 後直接引導到謝謝回應等等。
Complete Binary Tree	停車場停車時/當地下一樓停滿才開放地下二樓等等	當最上面的樓層都填滿後才往樓下停，符合時間成本及 complete binary tree 的概念，要先填滿最上層的。
Binary Search Tree	字典/以筆畫為排序時	先找隨機的頁碼，筆畫比欲找的小就往前翻，比欲找的大就往後翻，找到該字後，詞彙也相同，再比較詞彙第二個字筆畫等等。
AVL Tree	電腦中快速搜尋資料時	快速搜尋資料時，以搜尋的時間成本為重點，因此必須為平衡的。(插入刪除時間較長)
Red-Black Tree	電腦排程任務 operating system	常常刪除及插入，因修正的成本也較 avl 低，時間上也因樹平衡而不會太大。
Max Heap	電腦裡決定要完成的優先順序，如 operating system	如若值越大時越緊急，則透過最大的值放前面可優先處理。
Min Heap	急診室中決定要先急救的對象	如若值越小時越緊急，則透過最小的值放前面可優先處理。

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?

BST 不常更新時，BST 查詢最快，人名直接比大小最迅速。

Frequent insertions and deletions? Why?

紅黑樹，更新成本低，也不會像 BST 在更新後可能變成不平衡影響時間及後續。

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?

sorted array + binary search，不更新不用用那麼麻煩的其他結構，只是搜尋還是 BS 最方便快速，使用 sorted array 從中間值取出當 root 及其他子樹建立平衡的即可。

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	2025/12/29 20:38	ChatGPT	BST->red black 的條件是甚麼阿
2	2025/12/29 20:38	ChatGPT	可以告訴我更多紅黑樹的定義嗎 我覺得我有少理解的
3	2025/12/30 01:22	ChatGPT	可以幫我找有關 Complete binary tree 建立的網站嗎我找不到
4	2025/12/30 02:40	ChatGPT	可以解釋給我聽為什麼 AVL 在快速搜尋中是常用的嗎 我不太會解釋
5	2025/12/30 03:29	ChatGPT	可以幫我查看我有哪裡寫錯或可以加強的地方嗎

You may extend this table as needed.