

Practical 10 – Advanced Trees

Thi Van Anh DUONG Student ID: 90023112

Diploma of Information Technology, Curtin College

DSA1002: Data Structures and Algorithms

Coordinator: Khurram Hameed

7 August 2022

Student Declaration of Originality

<input checked="" type="checkbox"/>	This assignment is my own original work, and no part has been copied from another student's work or source, except where it is clearly attributed.
<input checked="" type="checkbox"/>	All facts/ideas/claims are from academic sources are paraphrased and cited/referenced correctly.
<input checked="" type="checkbox"/>	I have not previously submitted this work in any form for THIS or for any other unit; or published it elsewhere before.
<input checked="" type="checkbox"/>	No part of this work has been written for me by another person.
<input checked="" type="checkbox"/>	I recognise that should this declaration be found to be false, disciplinary action could be taken and penalties imposed in accordance with Curtin College policy.

Electronic Submission:

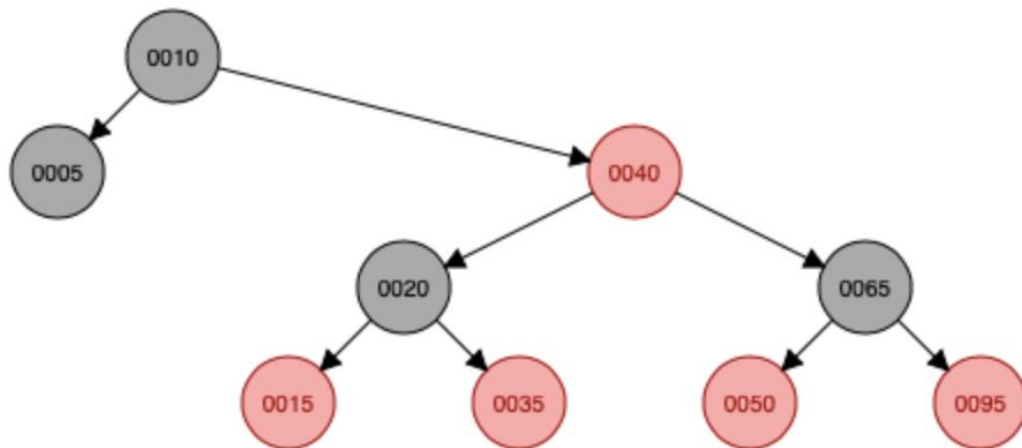
<input checked="" type="checkbox"/>	I accept that it is my responsibility to check that the submitted file is the correct file, is readable and has not been corrupted.
<input checked="" type="checkbox"/>	I acknowledge that a submitted file that is unable to be read cannot be marked and will be treated as a non-submission.
<input checked="" type="checkbox"/>	I hold a copy of this work if the original is damaged, and will retain a copy of the Turnitin receipt as evidence of submission.

ACTIVITY 1: RED-BLACK TREES

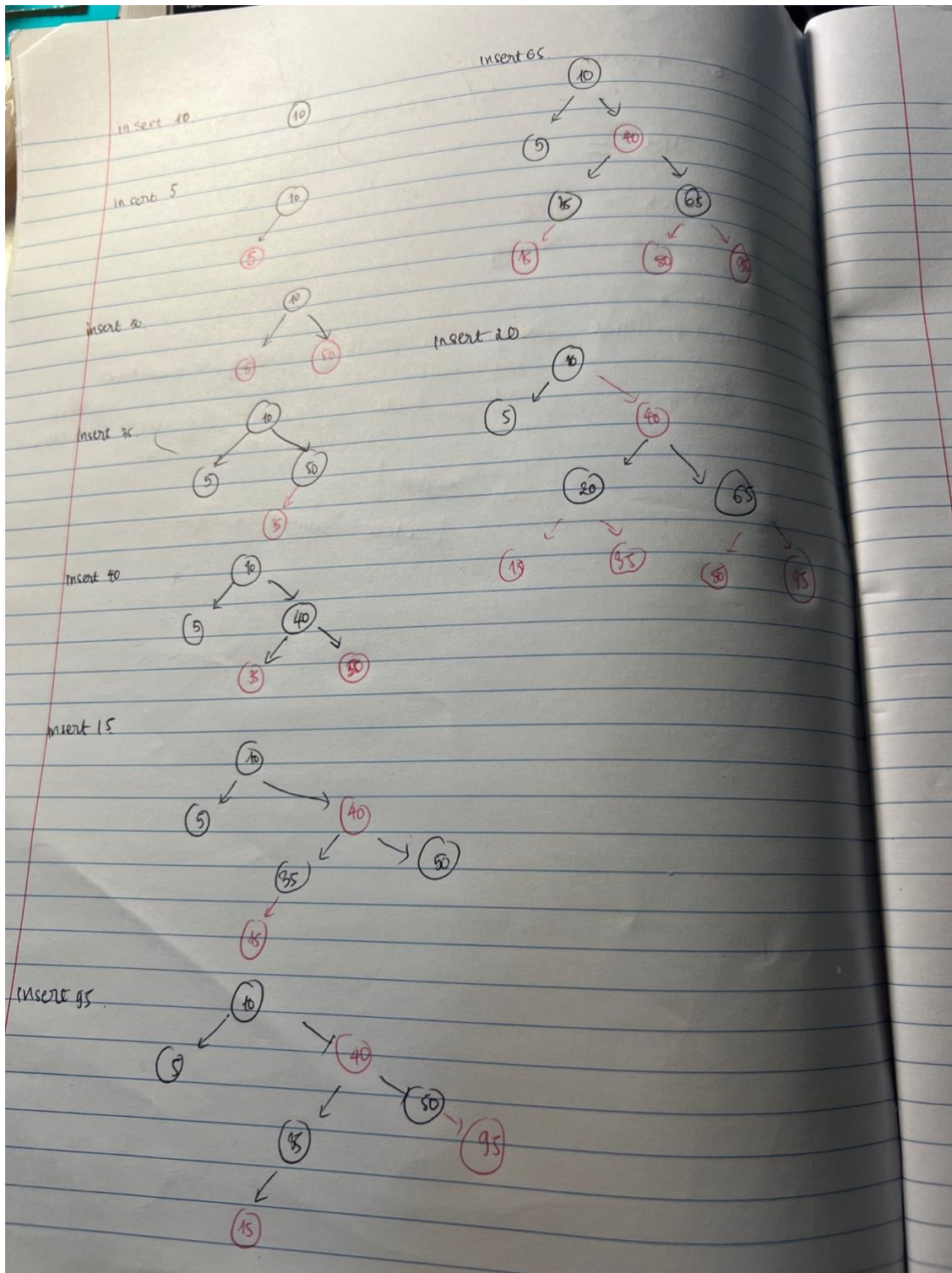
Tree 1:

(10, 5, 50, 35, 40, 15, 95, 65, 20)

App:



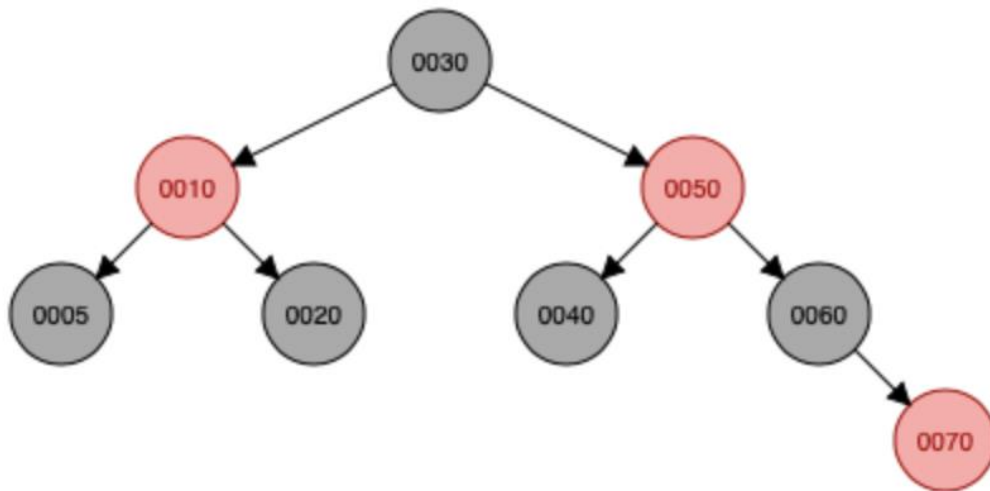
Manually:



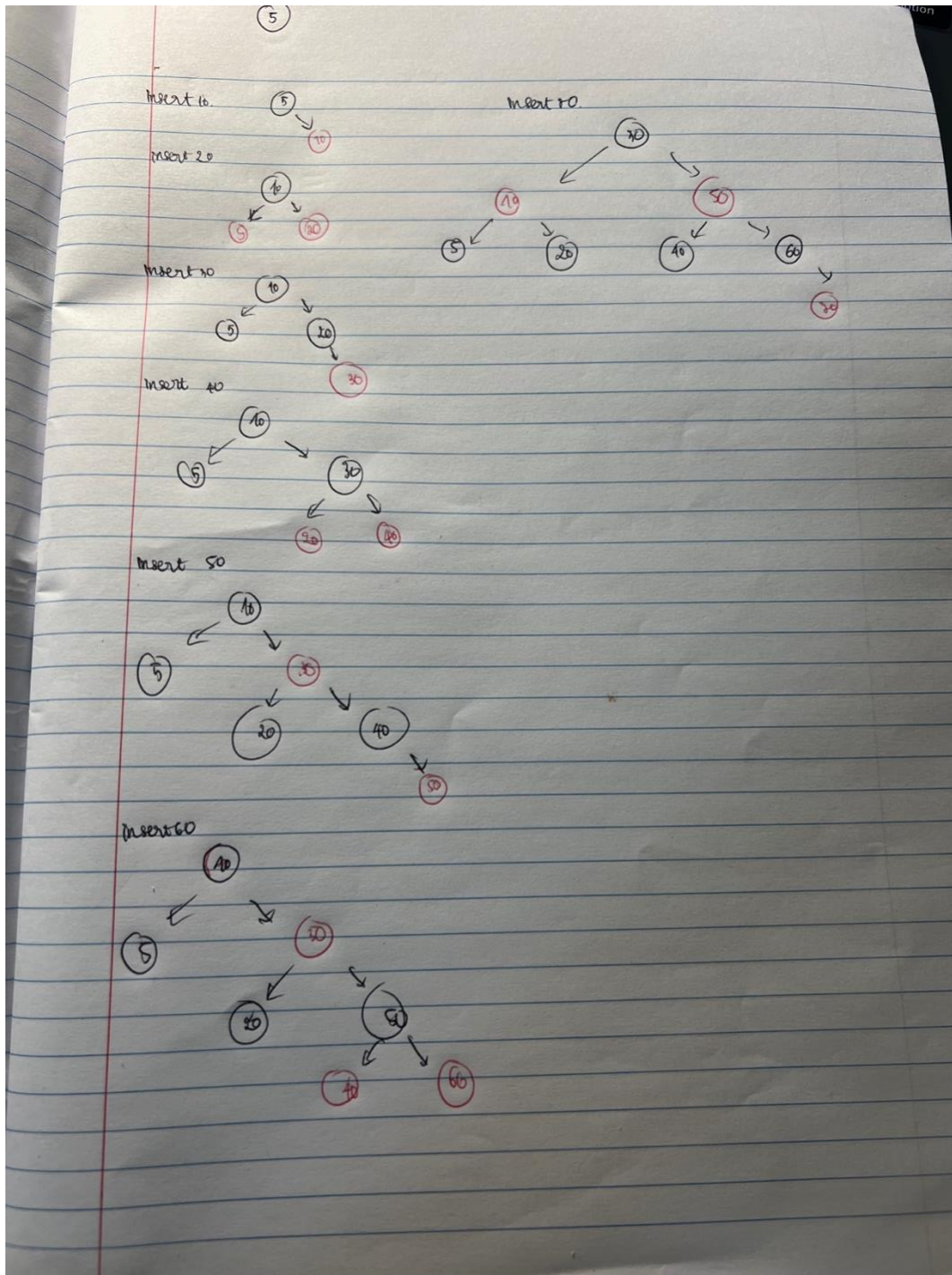
→ Manual draw looks the same as app.

(5, 10, 20, 30, 40, 50, 60, 70)

App:

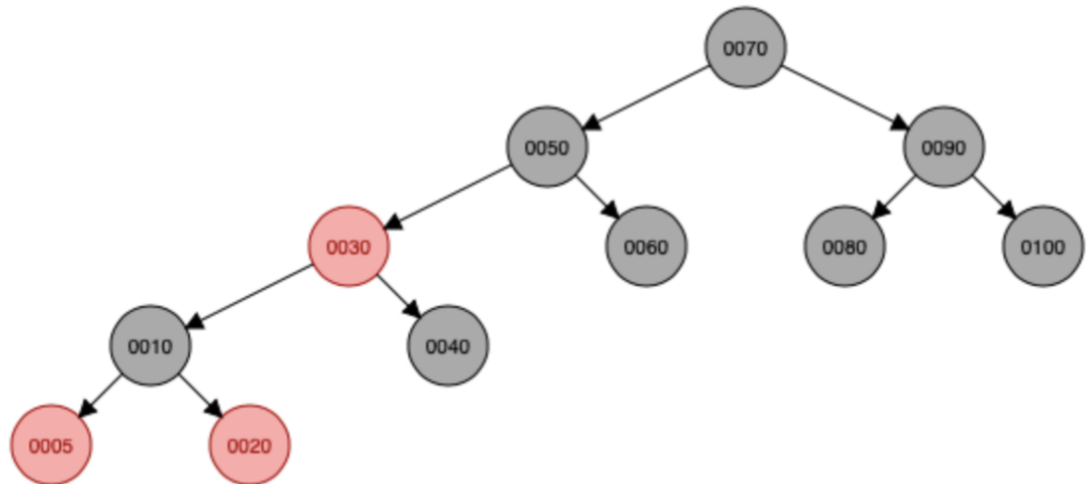


Manual:

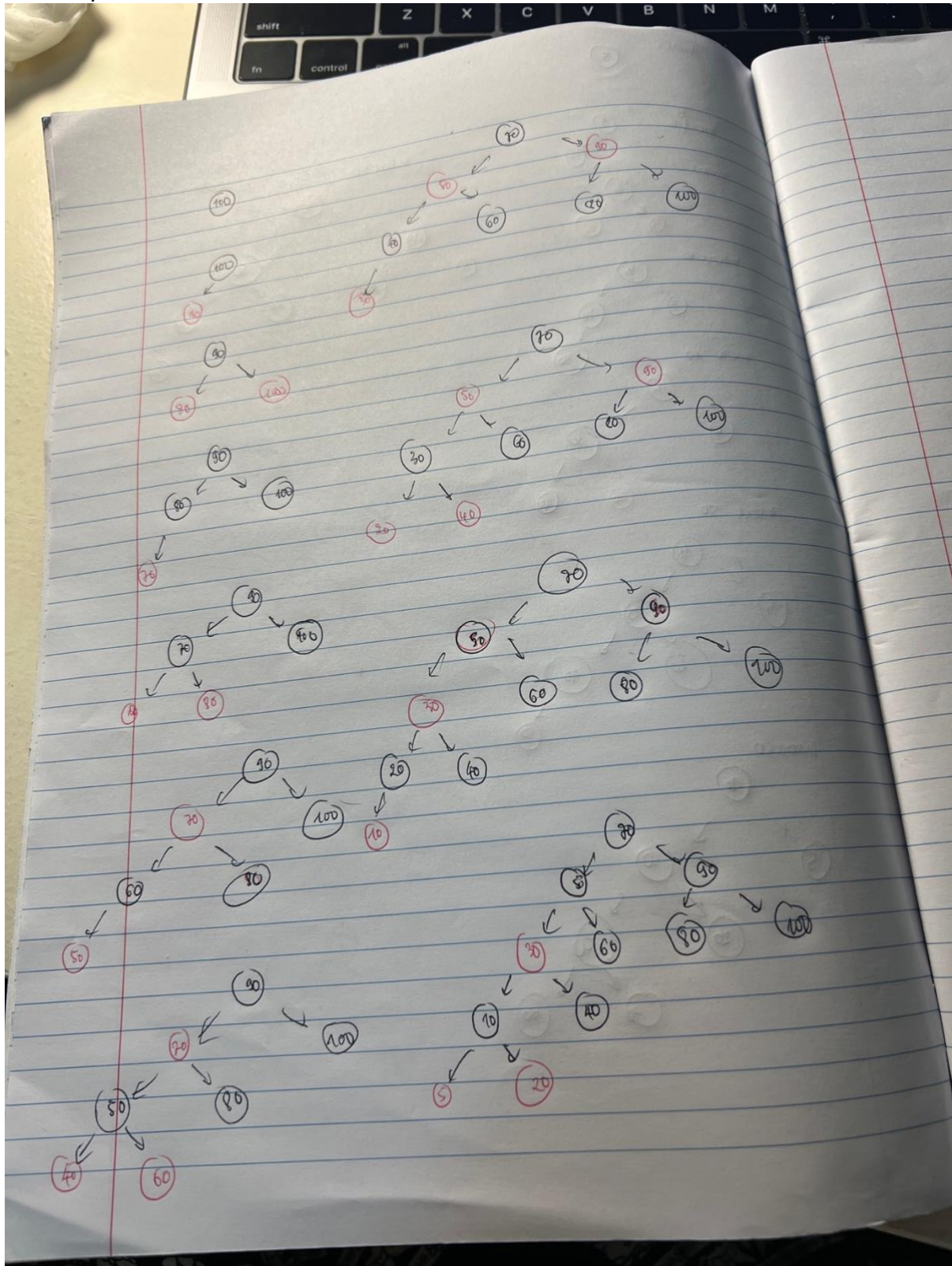


→ Manual draw looks the same as app.

(100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 5)



Manually:

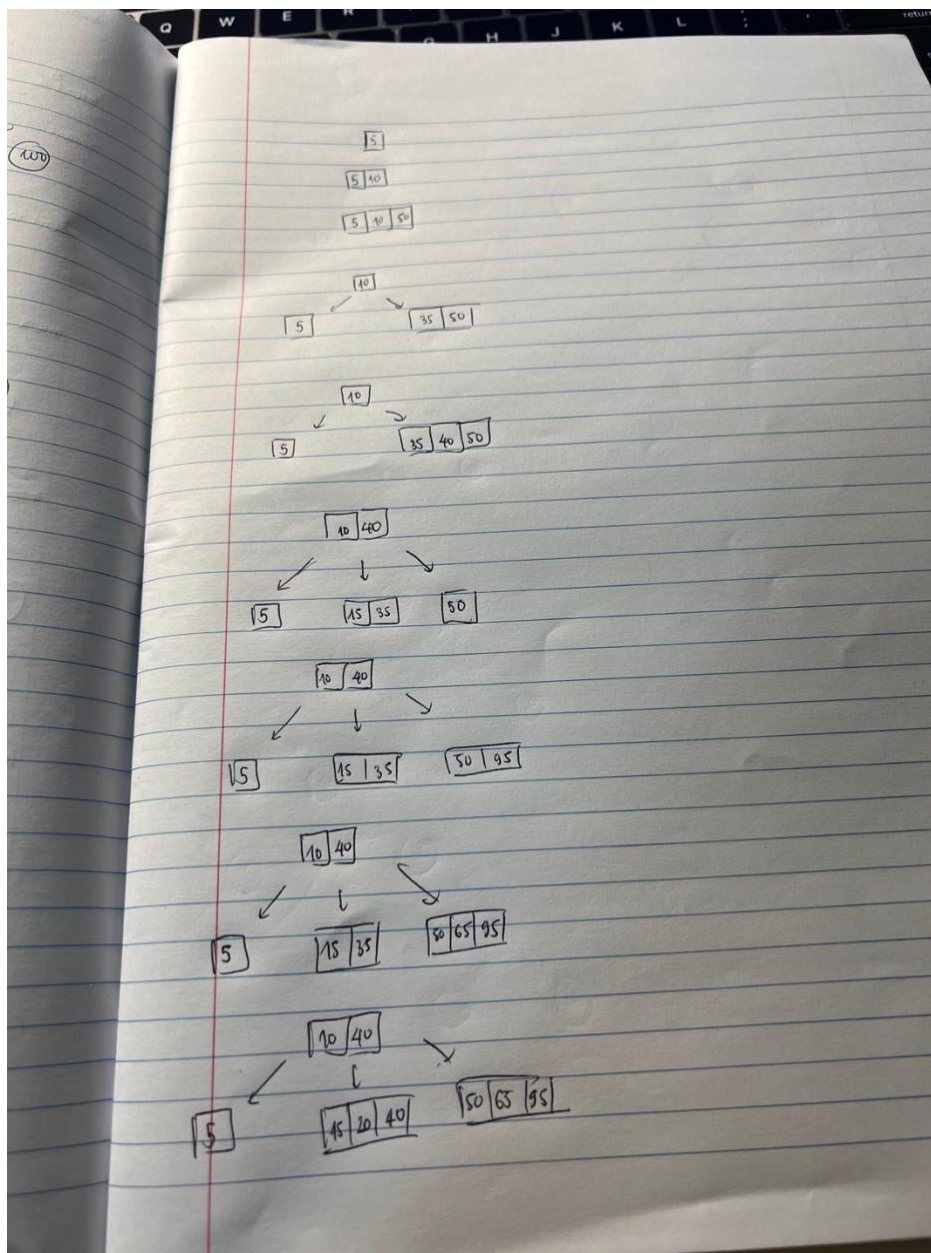


→ Manual draw looks the same as app.

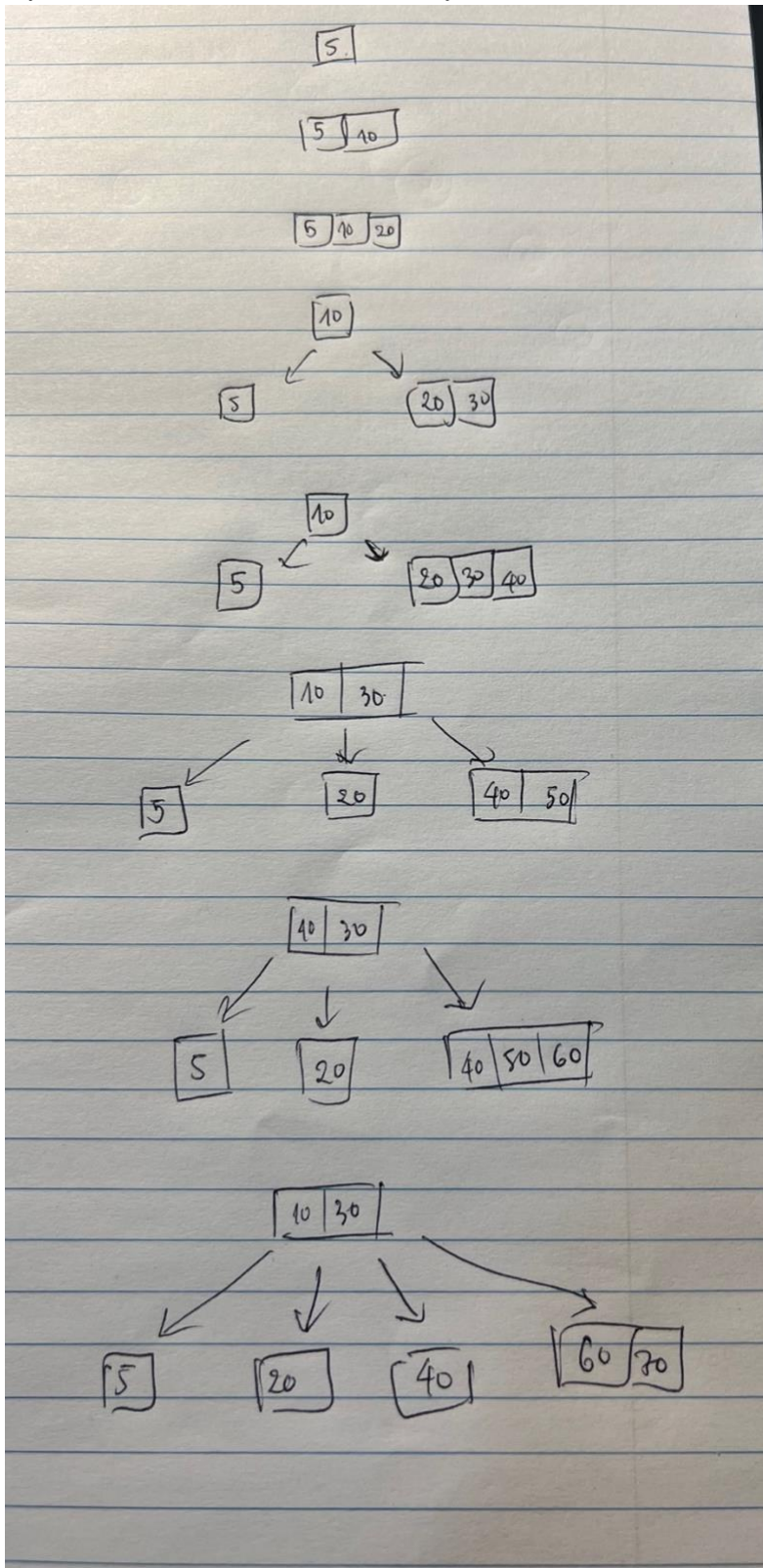
ACTIVITY 2: 2-3-4 TREES

I cannot open the link for 2-3-4 tree. So I will write down my work only.

(10, 5, 50, 35, 40, 15, 95, 65, 20)

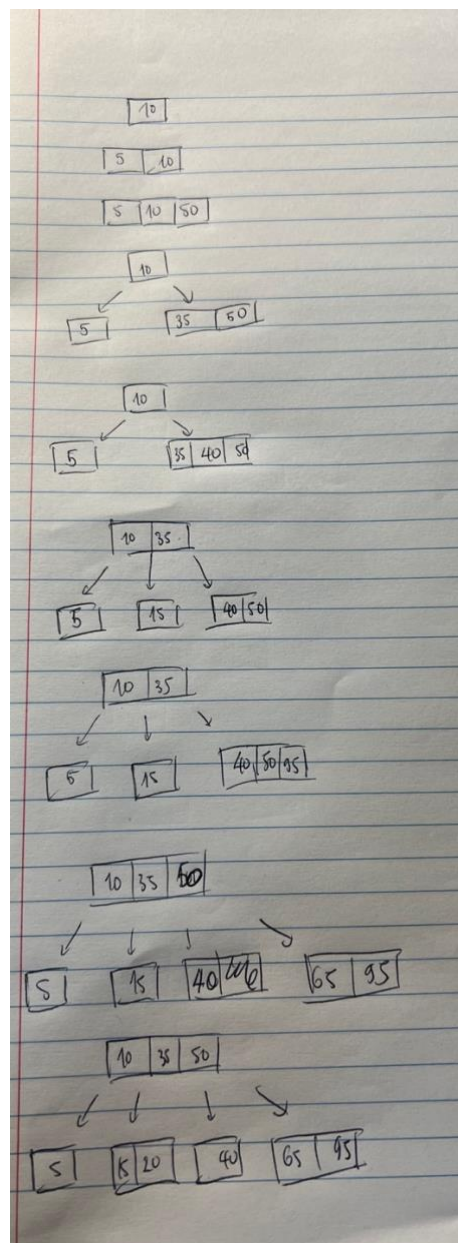


(5, 10, 20, 30, 40, 50, 60, 70)

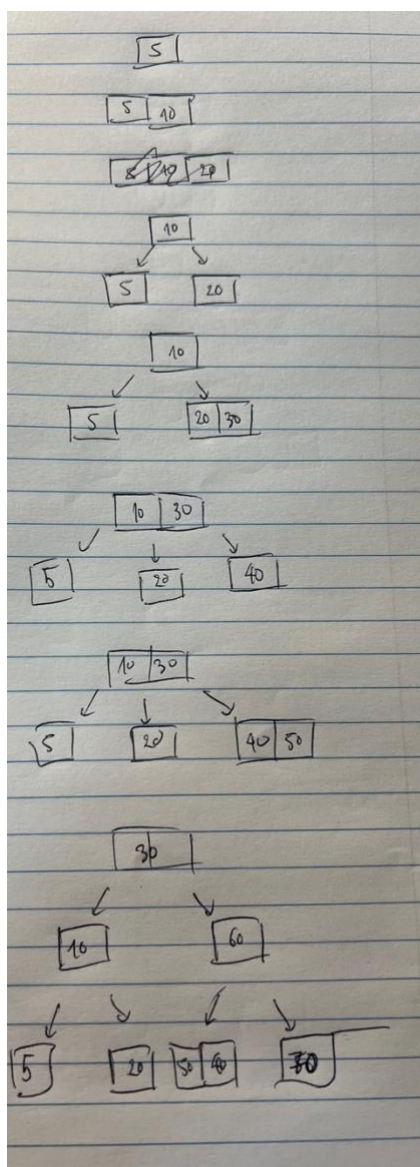
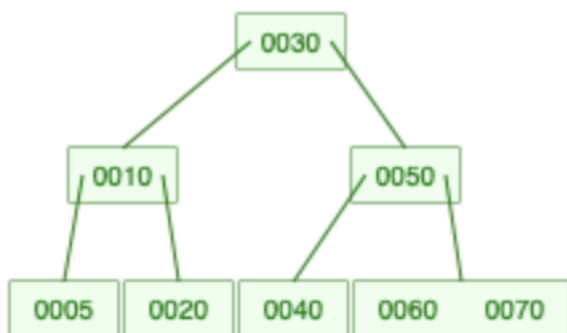


ACTIVITY 3: B-TREES

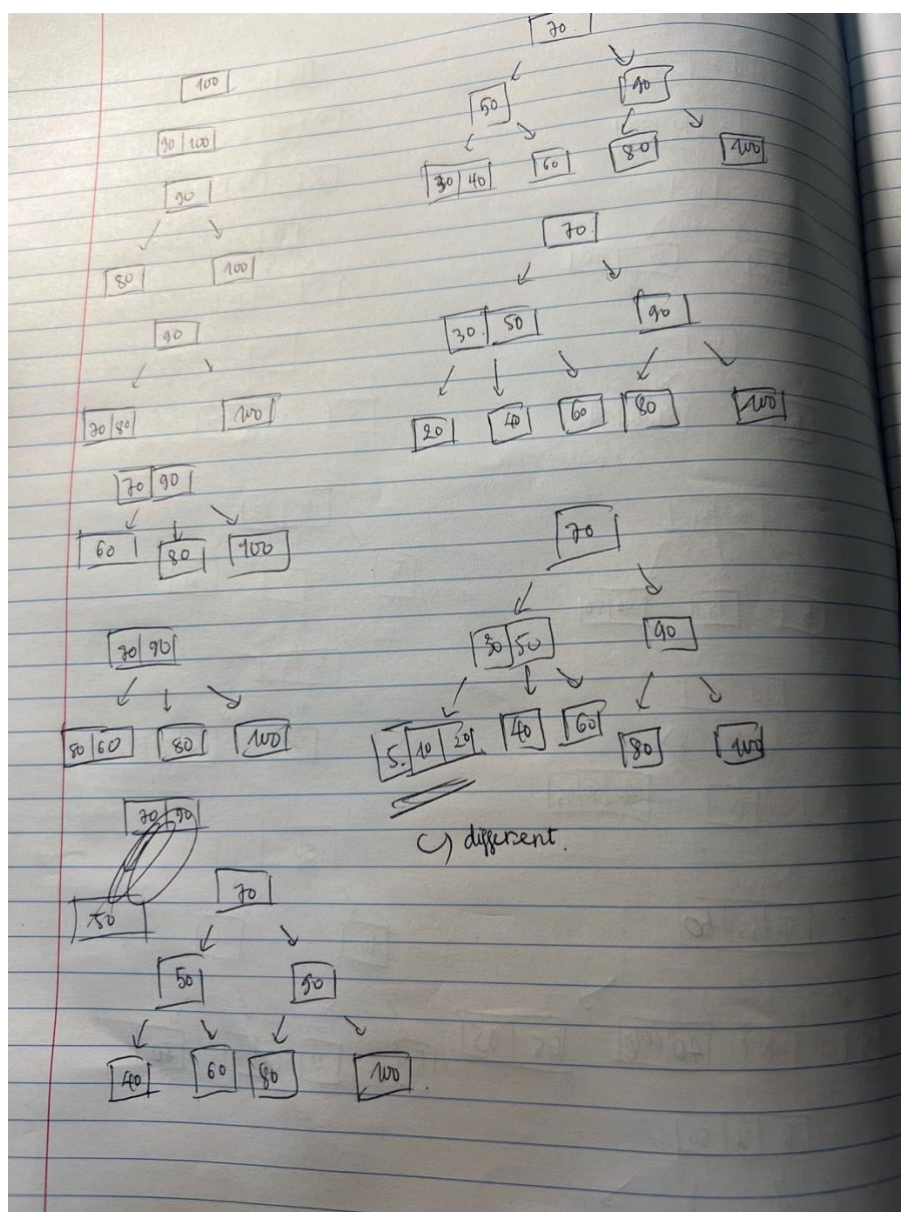
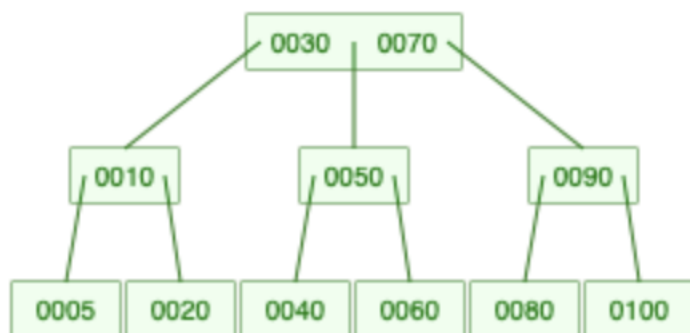
(10, 5, 50, 35, 40, 15, 95, 65, 20)



(5, 10, 20, 30, 40, 50, 60, 70)

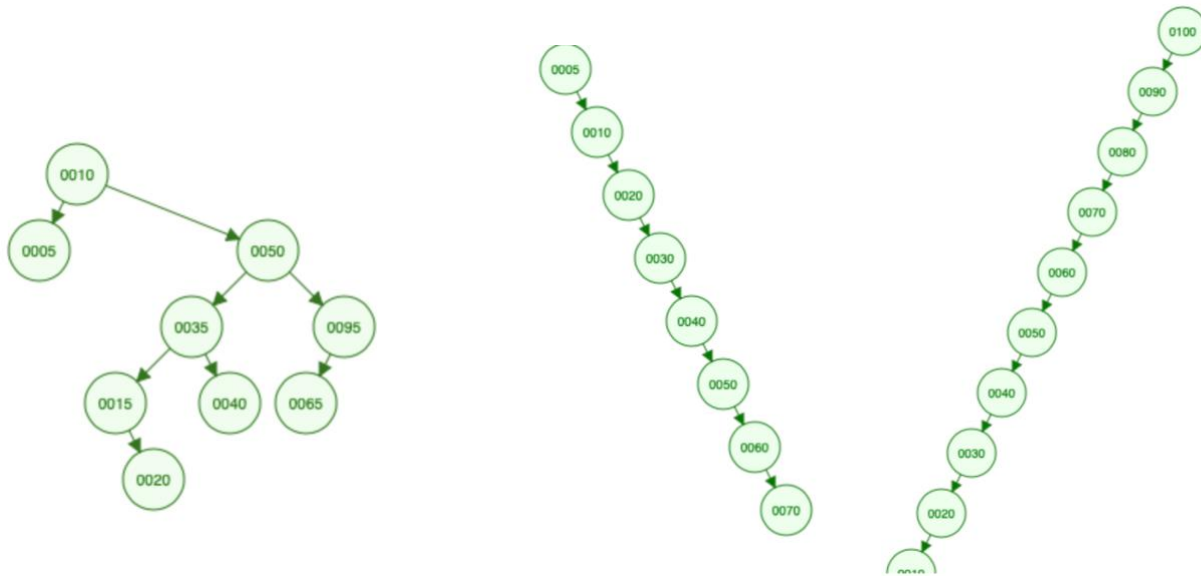


(100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 5)



ACTIVITY 4: REFLECTION

Binary Search Tree:



	Binary Search Tree	Red-Black Trees (1)	2-3-4 Trees (2)	B-Trees (3)
Height	Longer than (1), (2), (3)	Shorter than BST	Shorter than BST	Shorter than BST
Time Complexity	Insert: $O(N)$ Find: $O(N)$ Delete: $O(N)$	Insert: $O(\log N)$ Find: $O(\log N)$ Delete: $O(\log N)$ ➔ faster	Insert: $O(\log N)$ Find: $O(\log N)$ Delete: $O(\log N)$ ➔ faster	Insert: $O(\log N)$ Find: $O(\log N)$ Delete: $O(\log N)$ ➔ faster
Understandability	Easier to understand and implement	harder to understand and implement	harder to understand and implement	harder to understand and implement

In-order traversal:

Red-Black Trees:

- Check the left node first, then check the root and the right node.

2-3-4 Trees:

- Go from the root left, middle left, middle right and right node.

B-Trees:

- Traversal is also similar to Inorder traversal of Binary Tree. We start from the leftmost child, recursively print the leftmost child, then repeat the same process for the remaining children and keys. In the end, recursively print the rightmost child.