Write-up:

This week we worked on Binary search trees. This week would be my first time interacting with trees in general. I enjoyed this sorting algorithm, it’s interesting to see pre, in, and post order in action with unique data sets. Binary decision is relatively similar to if else statements, at least that is how I visualize that specific data structure. As traversing through binary trees, I found the three different methods are simple. Pre-order arranges nodes in sequence root, left, right. In-order is left root right. Post-order left right node. That is how I keep pattern simple to understand. For the last 2 questions it only made sense to use level order, starting index is at root continuing down each level from left to right.

1a. 16 34 35 38 39 41 44 45 55 63 64 65 72

1b. 45 38 34 16 35 41 39 44 65 63 55 64 72

1c. 16 35 34 39 44 41 38 55 64 63 72 65 45

1d. The height of the binary tree is 4. The nodes on level two are 38 and 65.

Shape

Description automatically generated with medium confidence

Shape

Description automatically generated with low confidence

2a. 48 – 7 % 2 / 24 \* 18 – 5 \* 2 + 12

2b. 48 7 2 % – 24 / 18 5 2 \* – 12 + \*

2c. 20

2d. 39.167

A picture containing graphical user interface

Description automatically generated

Level ordered

3a.

74 6 18 20 5

14 73 21 7 null 19 6 null 51 null null null 45

3b. 14 73 21 7 74 19 6 6 51 18 20 5 45

A picture containing logo

Description automatically generated

4.

