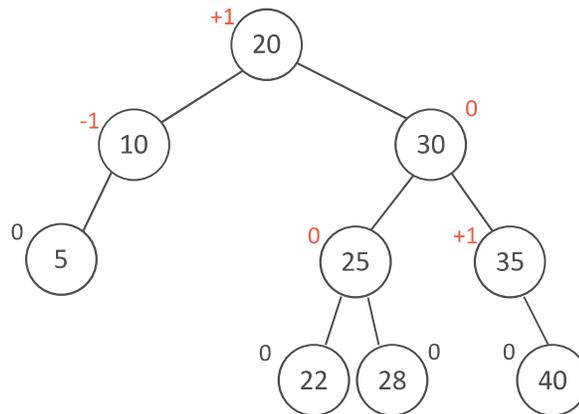


1. (12 pts.) AVL trees: Consider the AVL tree shown below.



- 1.1. Choose the correct balance for the node with key 10? [ -2 / -1 / 0 / +1 / +2].
- 1.2. Choose the correct balance for the node with key 20? [ -2 / -1 / 0 / +1 / +2].
- 1.3. Choose the correct balance for the node with key 25? [ -2 / -1 / 0 / +1 / +2].
- 1.4. Choose the correct balance for the node with key 30? [ -2 / -1 / 0 / +1 / +2].
- 1.5. Choose the correct balance for the node with key 35? [ -2 / -1 / 0 / +1 / +2].
- 1.6. Which node, if any, when removed would necessitate a rotation(s)? [ List the node's key or type "none" if none exists]. **5 or 10**
- 1.7. Show the process for **insert(29)** including where it is added, updates of any balance values, and the result of any necessary rotations. You can draw the tree at a few points in time during the insert process and need only draw the relevant parts of the tree (though you must show the relevant parts). Upload a PDF or image of your work.

<ul style="list-style-type: none"> <li>• Show correct insertion point of 29</li> <li>• Can earn partial credit if they show intermediate updates to parent balance values up to the root before the rotations (this is not necessary to get full credit, but if the rotation is wrong, you can give 1 point partial credit for these balance values)</li> </ul>	<ul style="list-style-type: none"> <li>• Correct zig-zag rotation on 20, 30, 25               <ul style="list-style-type: none"> <li>○ 25 at top</li> <li>○ 20 to left</li> <li>○ 30 to right</li> <li>○ 28 and 29 as left subtree of 30</li> <li>○ 22 as right subtree of 20</li> <li>○ Correct balance values</li> </ul> </li> </ul>