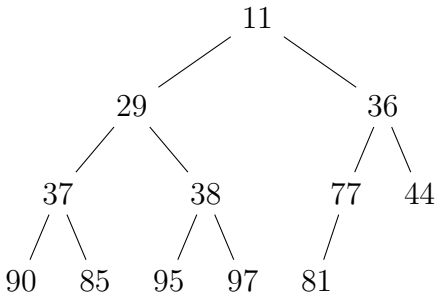
Name: _____

1. (1 pt.)
   - **Read all material carefully.**
   - Budget your **time**: 45 minutes, 45 pts. ⇒ 1 min./pt. avg.
   - You may refer to your **books**, papers, and notes during this test.
   - **Electronic books** are allowed *only when used as described in class.*
   - **No computer** or network access of any kind is allowed (or needed).
   - Write, and draw, **carefully**. Ambiguous or cryptic answers receive zero credit.
   - Use the definitions and **conventions** used in class and the textbook for notation, algorithmic options, etc.

   Write your name in the space provided above.

   Wait for the go signal before proceeding to next page.

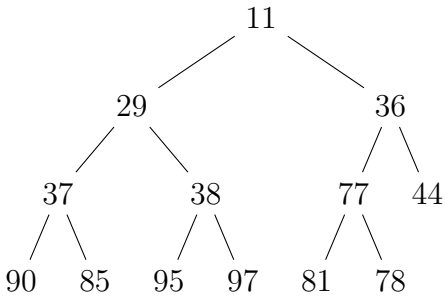
2. (10 pts.) Indicate how the key 22 is inserted into the following binary min-heap. Show all steps, as in the textbook.[1]



[1]Mark Allen Weiss, *Data Structures and Problem Solving Using Java*, 4th edition (Addison-Wesley, 2010), Figures 21.7 and 21.8.

[additional space for answering the earlier question]

3. (10 pts.) Indicate the heap that results from a *deleteMin* operation on the following binary min-heap. Show all steps, as in the textbook.[2]



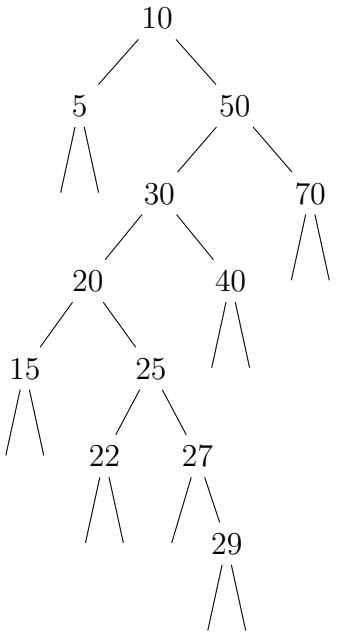
[2] *Idem*, Figures 21.10–21.12.

[additional space for answering the earlier question]

4. (10 pts.) Is the following tree a valid *bottom-up splay tree*?

   If not, explain your answer and provide a valid bottom-up splay tree that is as similar to it as possible. *Otherwise*, indicate the heights of the nodes with keys 10 and 20.

   In either case, depict the effect of a *find*(29) operation on the tree. Depict the intermediate states of the tree before and after every zig, zig-zig, and zig-zag rotation resulting from this *find* operation. Indicate where each rotation is applied by annotating the *root of the rotated* ***sub****tree* with an asterisk ($\star$).



[additional space for answering the earlier question]

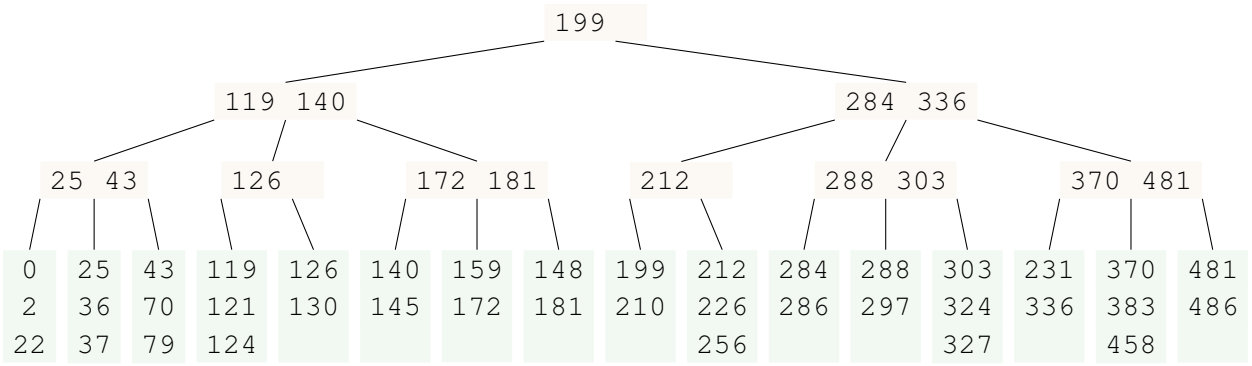
5. (14 pts.) Does the following figure depict a valid *B-tree with parameters* $M = 3$ *and* $L = 3$?

   If not, explain your answer and depict a valid tree that is as similar to it as possible. *Otherwise*, indicate *maximum* number of insertions that could be made to the tree without causing any node splits.

   In either case, depict the result of applying an *insert*(42) operation to it.

   Clearly describe the intermediate states of the tree before and after any node splitting operations (by redrawing the relevant portion of the tree for each such operation).

   You do *not* need to redraw the entire tree at each step.



[additional space for answering the earlier question]

[additional space for answering the earlier question]