

Name: _____

1. (1 pt.)

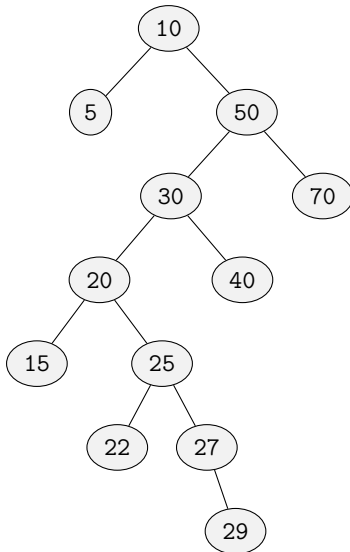
- **Read all material carefully.**
- *If in doubt whether something is allowed, ask, don't assume.*
- You may refer to your books, papers, and notes during this test.
- E-books may be used subject to the restrictions noted in class.
- No computer or network access of any kind is allowed (or needed).
- Write, and draw, carefully. Ambiguous or cryptic answers receive zero credit.
- Use class and textbook conventions for notation, algorithmic options, etc.
- There is an extra-credit question (marked with ★). It is harder than the rest.

Write your name and group ID (e.g., C3) in the space provided above. The group is for reference only; all work on this quiz is individual work.

2. (9 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 subtree* with an asterisk (*).



[additional space for answering the earlier question]

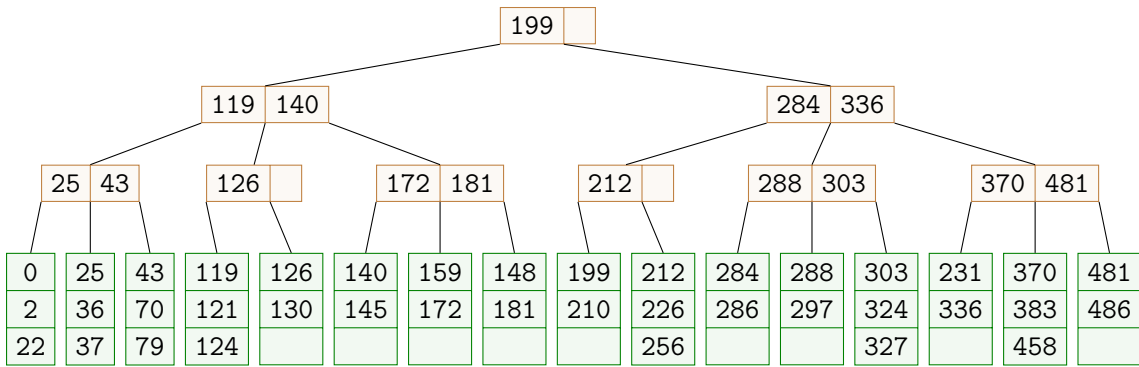
3. (10 pts.) Does the following figure depict a valid *B-tree with parameters $M = 3$ and $L = 3$* ?

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.

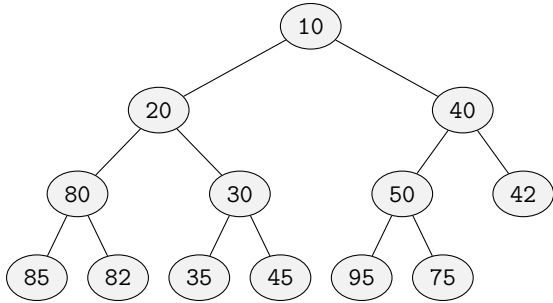


[additional space for answering the earlier question]

4. (10 pts.) Does the following figure depict a valid *binary heap (min-heap)*? If not, depict the result of applying the *heapify* algorithm to it. Otherwise, depict the result of applying a *deleteMin* operation to it.

Depict the intermediate states of the heap in either case, indicating clearly the nodes that get swapped, if any.

Also depict the final state of the heap in *implicit (i.e., array)* form.



[additional space for answering the earlier question]

5. (10 ★ pts.) Is the following tree a valid *bottom-up splay tree*? If not, explain why, and replace it with a valid bottom-up splay tree that is as similar to it as possible and continue:

