

Name: \_\_\_\_\_

1. (1 pt.)

- **Read all material carefully.**
- You may refer to your books, papers, and notes during this test.
- 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 conventions used in class and the textbook for notation, algorithmic options, etc.

Write your name in the space provided above.

2. (12 pts.) Depict the action of *in-place heapsort* on the following array. Depict **both** the state of the array (in linear form) and the max-heap it encodes (in tree form), using a format similar to Figures 21.25–21.27 (pp. 824–825) in the textbook<sup>1</sup>:

- **after each** *percDown* operation in the first (*buildHeap*) phase **and**
- **after each** *deleteMax* operation in the second phase.

array a[i]:	84	27	13	90	46	99	51	31	73	26
indices i:	0	1	2	3	4	5	6	7	8	9

---

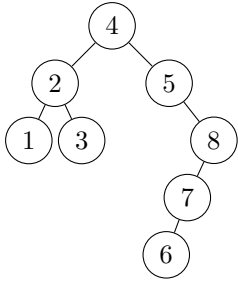
<sup>1</sup>Mark Allen Weiss, *Data Structures and Problem Solving Using Java*, 4th edition (Addison-Wesley, 2010).

[additional space for answering the earlier question]

[additional space for answering the earlier question]

3. (10 pts.) Depict the transformations to the following *top-down splay tree* in response to the access pattern 1, 2, 3, 4, 5, 6, 7, 8. For **each access**:

- Adorn the node being accessed with an asterisk \*.
- Clearly label **each splay operation** noting its type, and depict the tree before and after the operation.



[additional space for answering the earlier question]

4. (7 pts.)

- (a) Depict **all** AA-trees containing the four keys: 1, 2, 3, 4.
- (b) Repeat the above for AA-trees with five keys: 1, 2, 3, 4, 5.
- (c) Briefly explain why the trees you depict are the only possibilities.