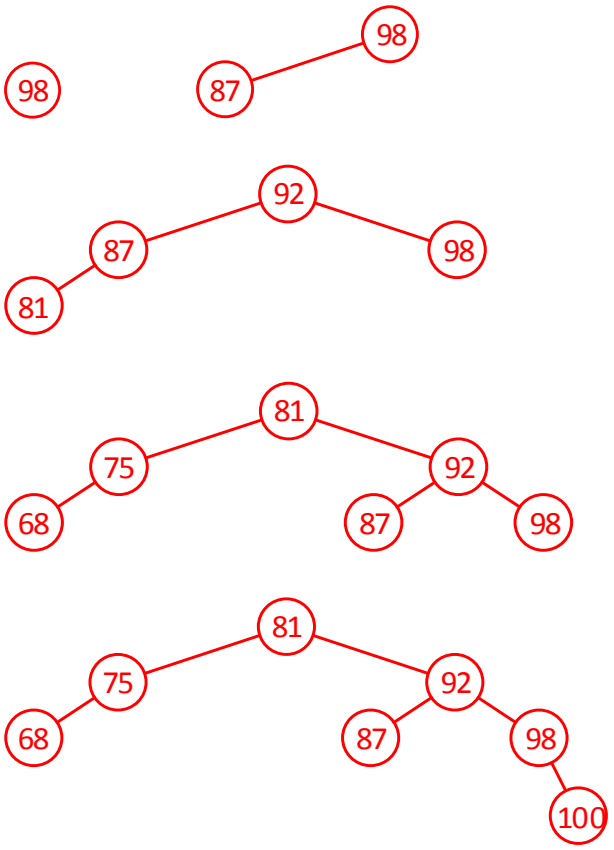
**3)** (10 pts) ALG (AVL Trees)

(a) (8 pts) An AVL tree stores the grades of the class (in between 1 and 100 inclusive). Create an AVL tree by inserting the following values into an initially empty AVL Tree in the order given: 98, 87, 92, 81, 75, 68, and 100. Show the state of the tree ***after*** each insertion and draw a box around each of these results.




**Grading: 1 pt per correct tree**
**1 pt if last tree is a valid AVL with all 7 elements**

(b) (2 pts) What is the fewest and most number of comparisons for looking for a valid grade that is ***not*** within this tree?

Fewest number of comparisons = **2 (e.g. 76 compares against 81 and 75 only – 1 pt, all or nothing)**

Most number of comparisons = **4 (e.g. 99 compares against 81, 92, 98, and 100– 1 pt, all or nothing)**

# Computer Science Foundation Exam

## August 31, 2019

## Section II A

## ALGORITHMS AND ANALYSIS TOOLS

## SOLUTION

**NO books, notes, or calculators may be used,**
**and you must work entirely on your own.**

| Question # | Max Pts | Category | Score |
|---|---|---|---|
| 1 | 10 | ANL | |
| 2 | 5 | ANL | |
| 3 | 10 | ANL | |
| TOTAL | 25 | | |

**You must do all 3 problems in this section of the exam.**

**Problems will be graded based on the completeness of the solution steps and <u>not</u> graded based on the answer alone. Credit cannot be given unless all work is shown and is readable. Be complete, yet concise, and above all <u>be neat</u>. For each coding question, assume that all of the necessary includes (stdlib.h, stdio.h, math.h, string.h) for that particular question have been made.**