CMPT 280: Intermediate Data Structures and Algorithms

Practice Midterm
Prepared by Mark Eramian

Disclaimer

These questions are meant to be representative of the common kinds of questions that might get asked on a CMPT 280 midterm examination. It is not an exhaustive sampling.

You should not conclude any of the following:

1. that these are the **only** concepts/topics/data structures that will be covered by the exam (ALL course materials in topics up to and including AVL trees are fair game for the exam!);
2. that these are the **only** types of questions that may be asked;
3. that the length of this exam will be similar to that of your actual midterm (this exam is approximately the length of a 45 minute exam; yours will be 90 minutes).

1. (1 point) If you needed a list where elements are very frequently added and removed from the end of the list, which of the following design rationales is the most sound?
   - A. We should use an array-based list where the tail of the list always is stored at index 0 of the array because that will mean faster access to the end of the list.
   - B. We should use a singly-linked list because it would take up less space over a doubly linked list.
   - C. We should use a doubly-linked list because it can be quickly read backwards from end to beginning.
   - [✓ D. We should use a doubly-linked list because insertion and deletion at the end of the list can be implemented much more efficiently than in a singly linked list.]

2. (1 point) Which statement(s) about cursors and iterators are true? (Choose one answer only! If more than one of A, B and C are true, choose one of the answers D through G.)
   - A. A cursor is internal to an ADT (abstract data structure); iterators are separate objects.
   - B. The user of an ADT can control how many cursors the ADT has.
   - C. Both cursors and iterators represent positions in a data structure.
   - D. A and B are both true.
   - E. B and C are both true.
   - [✓ F. A and C are both true.]
   - G. All of A, B, and C are true.

For questions 4 to 7 consider the two methods below:

```
public double a(double k[])
{
   double result = 1.0;
   for (int i = 1;  i < k.length; i = 2*i)
      result = result * p(i) * b(k);
   return result;
}

public  double b(double j[])
{
   double result = 0.0;
   for (int i = j.length-1; i >= 0; i = i - 1)
      result = result + j[i]/p(i);
   return result;
}
```

3. (1 point) Suppose that the method `p` has a return type of `double` and requires time $O(r)$ time, where r is independent of i. If we let $n = j.length$, what is the time complexity of method `b` in the worst case?
   A. $O(n)$ [✓ B. $O(nr)$] C. $O(r)$ D. $O(n^2)$ E. $O(r \log n)$

4. (1 point) Again suppose that the method `p` has a return type of `double` and requires time $O(r)$ time, where r is independent of i. If we let $m = k.length$, what is the time complexity of method `a` in the worst case?
   A. $O(mr)$ B. $O(r \log m)$ [✓ C. $O(mr \log m)$] D. $O(r^2 \log m)$ E. $O(mr^2)$

5. (1 point) Which line of the above code (line numbers are shown on the right) is an appropriate active operation for method `b`?

   A. Line 11   B. Line 12   C. line 13   D. Line 14   E. the of the above

6. (1 point) If we let $m = k.length$, exactly how many times is line 4 executed?

   A. $\lceil \log m \rceil$ B. m C. $m\lceil \log m \rceil$ D. m^2 E. 2^m

7. (1 point) Recall the formal ADT specification that we studied. Suppose set I is defined to be all instances of some kind of container (it doesn't matter what kind), G is the set of all elements that can be inside a container from C, and N_0 are the non-negative integers. Which of the following best reflects the information conveyed by the signature

$$C.\text{doSomething}(x, y, g) : N_0 \times N_0 \times G \to C$$

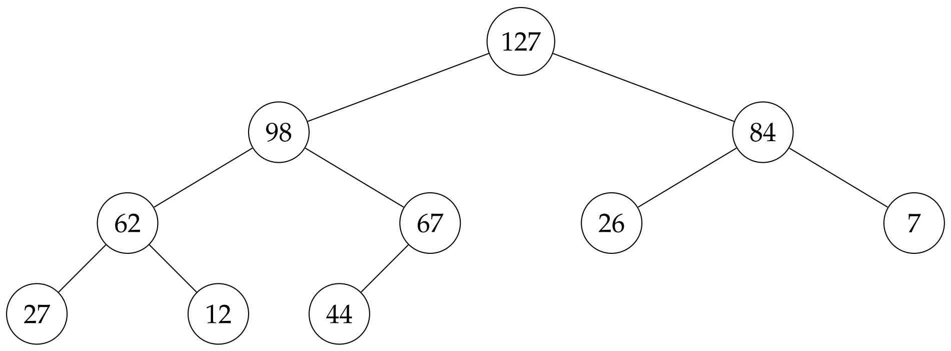
   A. doSomething is an operation that takes two non-negative integers, and an element as parameters, and alters the state of a container somehow.

   B. doSomething is an operation that sets the element at coordinate (x, y) to g.

   C. Same as in part A, but additionally, doSomething cannot be applied to all elements of C.

   D. doSomething should be implemented using a two-dimensional array.

For questions 11 through 13, consider the following **binary heap**:



8. (1 point) If the element at the top of the heap is deleted, how many element swaps will occur while restoring the heap property, not counting the initial replacement of the root element?

   A. 0   B. 1   C. 2   D. 3   E. 4

9. (1 point) If the above heap is stored in an array, at which array offset (index) will the element 67 appear assuming that the root element is at array offset 1?

   A. 4   B. 5   C. 6   D. 7   E. 8

10. (1 point) If the element 42 is added to the heap, how many swaps will occur while restoring the heap property?

    A. 0   B. 1   C. 2   D. 3   E. 4

11. (1 point) Which of the following is a correct declaration of a linked list of linked lists of Integers?

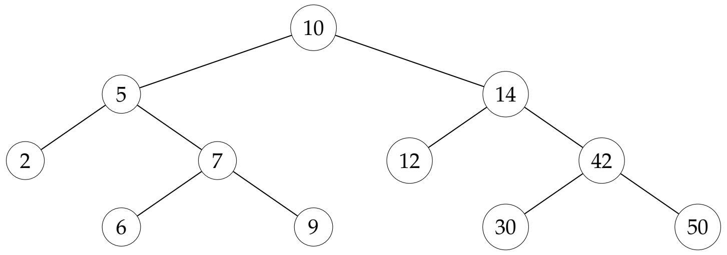
    A. `LinkedList280<Integer>[] listOfLists;`

    B. `LinkedList280< LinkedList280<Integer> >[] listOfLists;`

C. `LinkedList280< LinkedList280<Integer> > listOfLists;`

D. `LinkedList280< LinkedList280<Integer>[]> listOfLists;`

For questions 15 to 17, consider the following tree, and list of node traversals.



Traversal #1: 10 5 2 7 6 9 14 12 42 30 50

Traversal #2: 2 5 6 7 9 10 12 14 30 42 50

Traversal #3: 10 14 5 12 7 42 2 6 30 9 50

12. (1 point) Which of the above traversals is the in-order traversal of the tree?

    A. Traversal #1 B. Traversal #2 C. Traversal #3 D. None of the above.

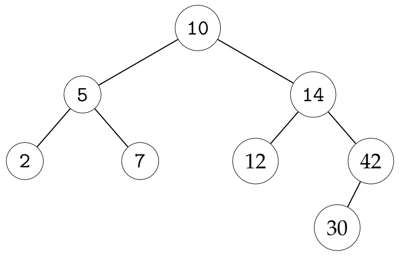
13. (1 point) Which of the above traversals is a breadth-first traversal of the tree?

    A. Traversal #1 B. Traversal #2 C. Traversal #3 D. None of the above.

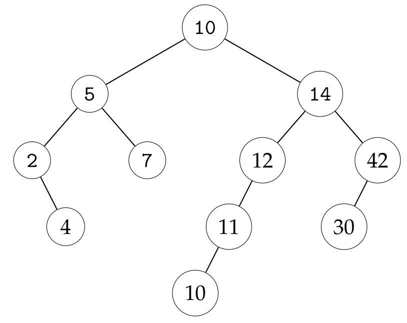
14. (1 point) Which of the above traversals is the post-order traversal of the tree?

    A. Traversal #1 B. Traversal #2 C. Traversal #3 D. None of the above.

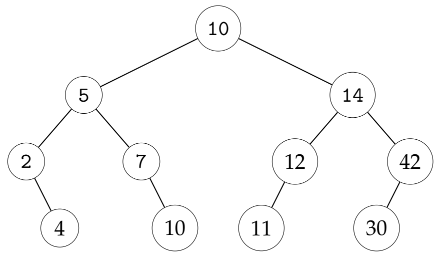
For questions 18 and 19, consider the following **ordered binary tree** (binary search tree).



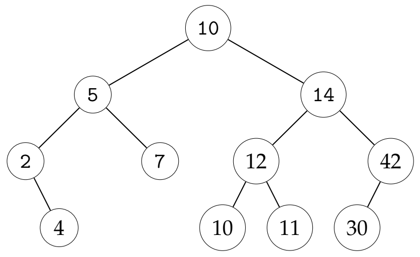
15. (1 point) Which of the following trees would result from the successive insertion of the elements 11, 4, and 10 (in that order) into the above ordered binary tree?



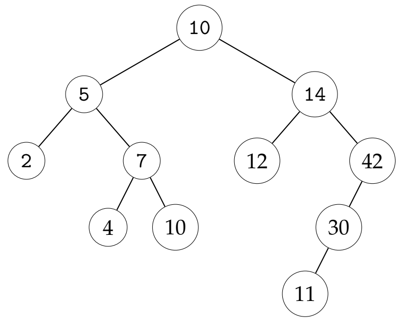
A



B



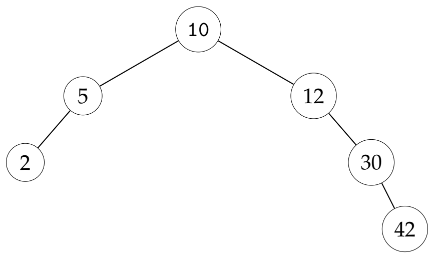
C



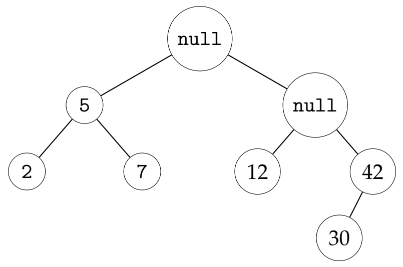
D

Solution: The correct answer is A because we adopted the convention of inserting equal elements to the right.

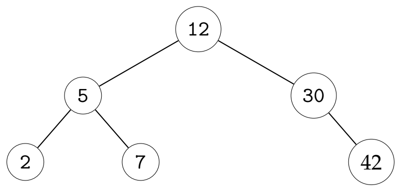
16. (1 point) Which of the following trees would result from the successive deletion of the elements 14 and 10 (in that order) from the above ordered binary tree?



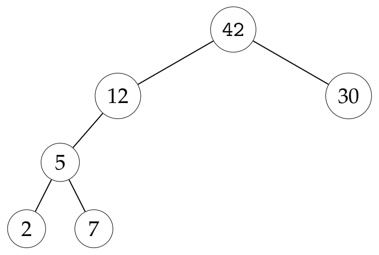
A



B



C



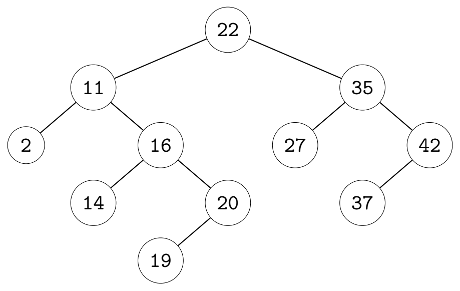
D

Solution: The correct answer is C.

17. (1 point) The worst-case time complexity of the insertion and deletion algorithms for **ordered binary trees** is:

A. $O(n)$ B. $O(n \log n)$ C. $O(\log n)$ D. $O(n^2)$ E. None of the above.

18. (1 point) Consider the following **AVL Tree**.



What type of imbalance does this AVL tree exhibit?

A. LL imbalance

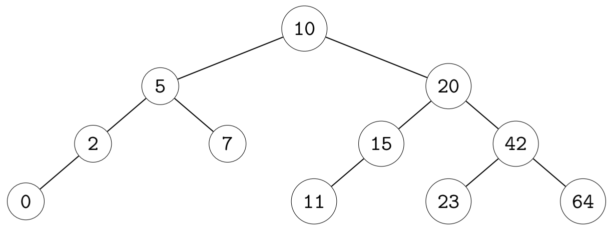
B. LR imbalance

C. RL imbalance

D. RR imbalance

E. No imbalance (maximum imbalance is 1)

19. (1 point) Consider the following **AVL tree**:



Supposing we insert the element 1, what type of rotation (if any) must be performed to restore the AVL property?

A. Right rotation

B. Double right rotation

C. Left rotation

D. Double left rotation

E. No rotation necessary.

20. (1 point) A double left rotation is:

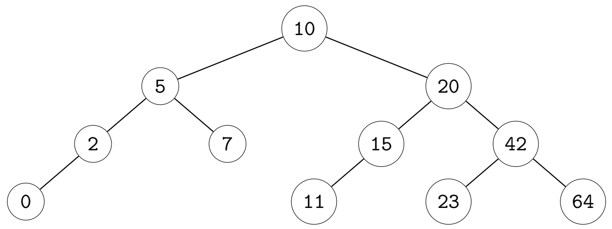
A. A right-rotation of the right subtree of the critical node, followed by a left rotation of the critical node.

B. A right-rotation of the right subtree of the critical node, followed by a right rotation of the critical node.

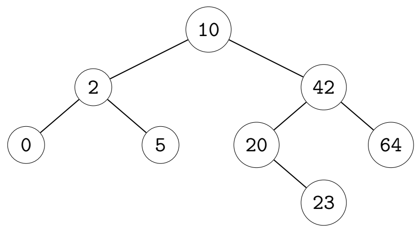
C. A left rotation of the left subtree of the critical node, followed by a left rotation of the critical node.

D. A right-rotation of the left subtree of the critical node followed by a left rotation of the critical node.

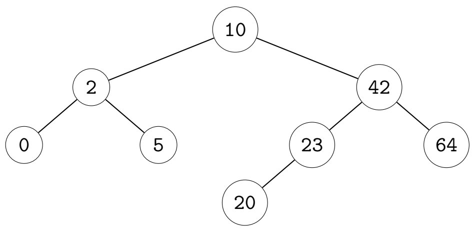
21. (1 point) Consider the following **AVL tree**:



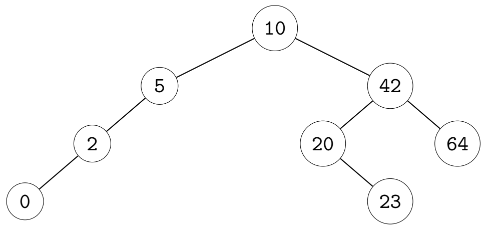
Which of the following trees results from the successive deletion (including any and all necessary rotations) of the elements 7, 15, and 11 (in that order)?



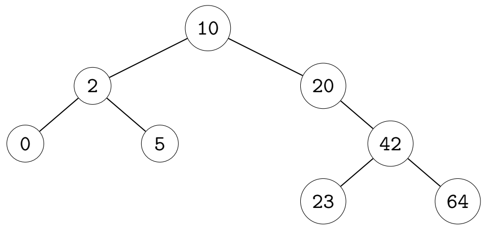
A



B



C



D

Solution: The correct answer is A.