Milwaukee School of Engineering

Electrical Engineering and Computer Science Department

CS-2852 – Test 2 – Dr. Durant

Thursday 24 April 2014

No notes, calculators, or other reference materials may be used.

Good luck!

Name: Answers

Page 2: (20 points) ______

Page 3: (20 points) ______

Page 4: (25 points) ______

Page 5: (20 points) ______

Page 6: (15 points) ______

Total: (100 points) ______

:47 :00 = 13m

1. (10 points) ***Illustrate*** what the stack for evaluating the postfix expression “$\underbrace{\underbrace{5\ 6\ +}_{11}\ 2\ *}_{22}$” looks like ***both*** immediately before processing ‘+’ ***and*** immediately after processing ‘*’.

↓

top → $\boxed{6}$

$\boxed{5}$

↓

$\boxed{22}$

2. (10 points) “$\overbrace{9\ \overbrace{2\ 3\ +}^{5}\ *}^{45}$ - 6” is not a valid postfix expression. ***Explain*** why it is invalid in terms what would happen to the stack during evaluation.

The stack would be empty when - requests its second operand.

$\boxed{45}$ ← can't apply -, only 1 of 2 needed operands

3. (5 points) "Stacks can be efficiently implemented using either a ArrayList or a LinkedList." ***Explain*** why this is true.

AL: bottom of stack @ [0], top moves up & down, no reshuffling of elements needed. Can grow when capacity exceeded.

LL: always modifying tail, adding or removing a node. no expensive traversals needed

4. (5 points) ***Define*** "adapter class."

a class that makes a class act like another class/interface.

From ③, we might adapt AL or LL to act as a stack.

5. (5 points) ***Explain*** how a "façade class" (or "façade") is ***different*** from an adapter class.

a class that provides a simplified interface to a complex object when the complexity is not needed.

E.g. a "SoundCard" may be complex, handling mono/stereo, multiple voices, etc. While a wavPlayer façade on it would provide a small # of 'simple' methods, hiding the complexity

6. (5 points) One can look at the element about to come out of a Queue using either the element() or the peek() method. ***Explain*** the ***difference*** between these methods.

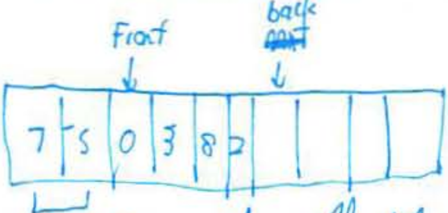
↓ null if empty

↓ exception if empty
needed if storing null in queue

7. (10 points) ***Explain*** why an ArrayList<E> is not an appropriate choice when implementing a ***pure*** queue interface.

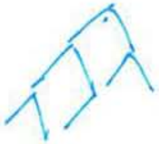
after removing items, the lower indexes would be forever unused

8. (15 points) ***Illustrate*** a circular queue of capacity 10 that has had the following elements added to it in order: 7, -5, 0, 3, 8, 2, followed by removing 2 elements. Be sure that your illustration ***makes clear*** which element is at the front of the queue.

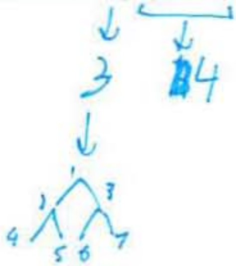


if objects, you'd null those 2 out to enable garbage collection

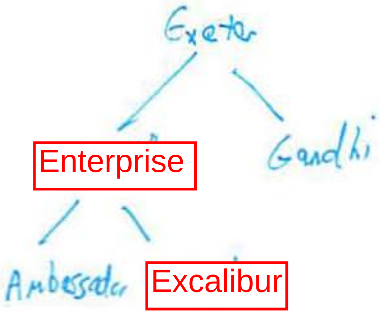
9. (5 points) ***Draw*** a binary tree that is ***complete*** but ***not full***.



10. (5 points) ***What*** are the minimum depths of binary search trees of size 7, 8, and 9? (3 answers)

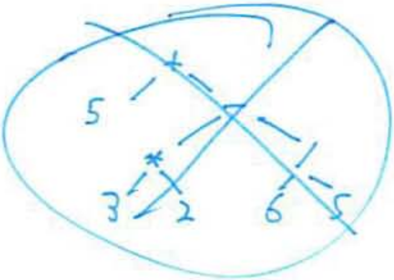


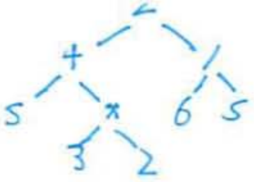
11. (10 points) ***Draw*** a binary search tree of ***minimum*** depth containing the following words ordered alphabetically: Exeter, Excalibur, Gandhi, Enterprise, Ambassador. There is more than one correct answer.



This is the only correct ~~full~~ complete tree, but a complete tree is not required here.

12. (10 points) ***Draw*** a binary expression tree for the infix expression "5 + 3 * 2 – 6 / 5" using standard rules of operator precedence.





13. (5 points) ***Write*** out the ***post-order*** traversal of your answer to the previous problem.

~~5 3 2 * 6 5 / – +~~

5 3 2 * + 6 5 / –