CPS305 Midterm Test Fall 2010

(24 marks, 5 pages)

First Name: ***Last Name.***

Student ID: ***Section or Lab Day/time***

Signature: _____

For questions 1-3, **what kind of recursion** does the given algorithm use? Circle the best answer for each. (1 mark each)

(a) tail recursion
(b) edges-and-center recursion
(c) last-and-all-but-last recursion
(d) division-in-halves recursion
(e) none of the above

where

- **A** is an array, and **x, y** are valid indexes of A, and **x <= y**
- **Join(K,M)** makes a list from item(s) in K followed by item(s) in M
- **C(x)** is the usual math ceiling function $\lceil x \rceil$, (if x is a whole number, it returns x; if x is a fraction it returns the next highest whole number.)

1. **algorithm F**: (a) (b) (c) (d) (e)

```
F(A, x, y)
    if (x == y) return 1
    if (x < y)   return  x *  F(A,x+1, y)
```

2. **algorithm G**: (a) (b) (c) (d) (e)

```
G(A, x, y, n)
    if (x == y)  return n     //n is irrelevant to the type of recursion this is
    if ( x < y)   return G(A, x+1, y, n+1)
```

3. **algorithm H**: (a) (b) (c) (d) (e)

```
H(A, x, y)
    if (x == y)  return A[x]
    if (x < y)    return Join(  H(A, x + C((y-x)/2), y),  H(A, x, x + C((y-x)/2) –1) )
```

For questions 4-8, circle the best answer. (1 mark each)

4. Which of the following is a **primitive data type** in **C**:
   a. array
   b. char
   c. complex number
   d. more than one of a, b, c
   e. none of a, b, c

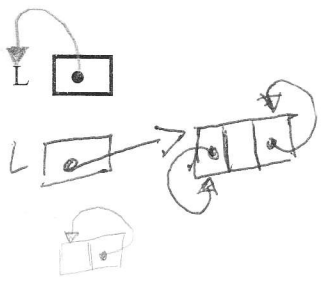
5. An **ADT**:
   a. is made up of operations and a data structure (object)
   b. uses the principle of information-hiding
   c. can have many lower-level representations
   d. more than one of a, b, c
   e. none of a, b, c

6. We used the technique of **unrolling** to:
   a. help us write the base case for a recursive function
   b. help us produce the annotations on call trees
   c. help us solve recurrence relations
   d. none of the above

7. An **iterative** algorithm to **multiply** together **two nxn matricies** will have **time** complexity
   a. O(n)
   b. $O(n^2)$
   c. $O(n^3)$
   d. $O(n^4)$
   e. none of the above

8. A List of n integers is implemented 2 ways: (1) an array of MAX items, (2) a linked list with standard nodes/pointers. Assume integers and pointers use equal storage, and that both implementations keep an integer "count" of the current number of items in the list.
   For large MAX and n, it is more space-efficient to use the first (array) implementation:
   a. when n > (1/3 * MAX), approximately
   b. when n > (½ * MAX), approximately
   c. when n > (¾ * MAX), approximately
   d. when n > (2 * MAX), approximately

9. (2 marks) List L is **circular**, **doubly-linked** (two-way), with **header**. The drawing below shows L after "L=NULL;". Change the drawing to show empty list, L, after "Initialize(&L);"



10. (3 marks) For each of the following Computer Science applications, indicate whether it tends to be associated **more** with Stacks, or Queues. **Put a checkmark** in the appropriate box:

LIFO FIFO

Application	Stack?	Queue?
print spool		✓
backtracking	✓	
evaluating postfix expressions	✓	
level-order tree traversal	✓	
answering server requests		✓
implementing function calls	✓	

2.5

11. (3 marks)
    a. On the algorithm below, **circle** what you would count for **time complexity**
    b. Use a **recurrence relation** to find **time** complexity, in **Big-O** notation, for the following algorithm. Assume n is always an integer >=1

Factorial (n)

if (n == 1) return 1

else return (n * Factorial (n - 1))

1

$T(0) = 1$ .5 $T(1) = k$

$T(n) = k + T(n-1)$

$= 2k + T(n-2)$

$= 3k + T(n-3)$ .5

$\vdots$

$= nk + T(n-n)$

$= n + T(0)$

$= n + 1$

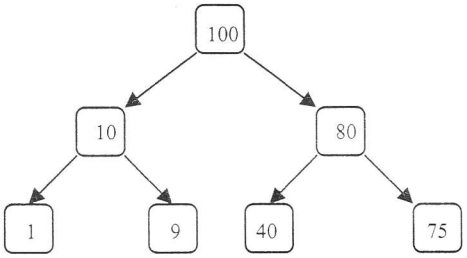
$= O(n)$

4.5

12. (2 marks) In the space below, give an in-order traversal of the following tree.

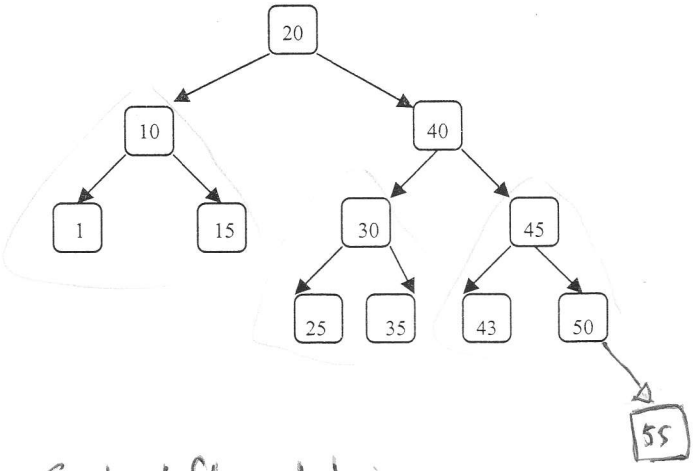
2

In-order Traversal: 1, 10, 9, 100, 40, 80, 75



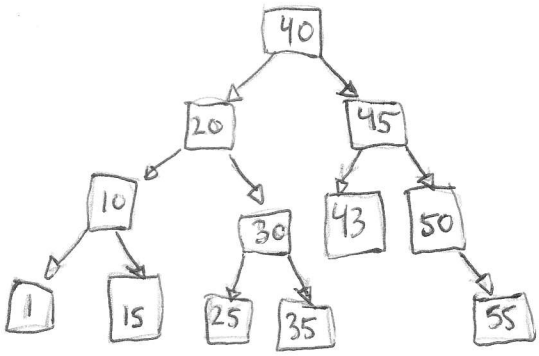
L K R

13. (2 marks) **Insert** key **55** into this **AVL** tree. Re-draw the new AVL tree.



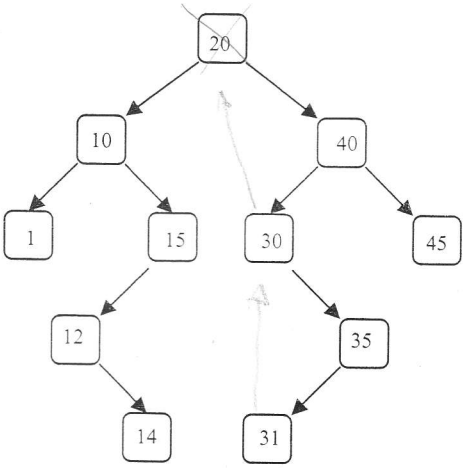
2

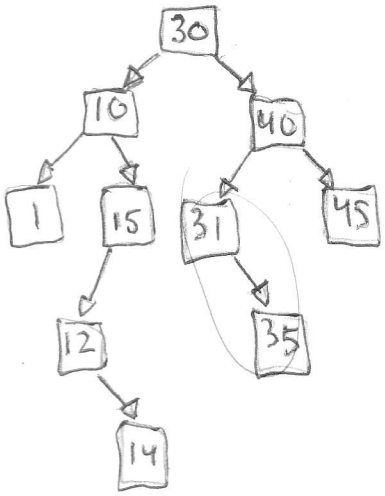
Single left rotation.



4

14. (2 marks) **Delete key 20** from the given **BST**. Re-draw the new BST.





15. (2 marks) Organize the following keys into a heap: 1, 2, 3, 4, 5, 6, 7, 8. **Draw only the final tree.**