CS235101 Data Structure Midterm Exam
Spring2015

(8%) [Algorithm]

1. (5%) Please sort the following terms according to its time complexity and fill the blanks.

$$a.O(n^2);\ b.O(n);\ c.O(n\log_2 n);\ d.O(\log_2 n);\ e.O(2^n).$$

$$O(1) < \_\_\_\_ < \_\_\_\_ < \_\_\_\_ < \_\_\_\_ < \_\_\_\_ < O(n!).$$

2. (3%) $g(n) = 7n^4 + 3^n + 100000 = O(?)$

(7%) [General Concept]

3.

(a) (2%) A queue follows a FIFO (first-in-first-out) rule, T or F?

(b) (2%) The time complexity of inserting an element into a linked list is O(1) if we already have a pointer points to its previous element, T or F?

(c) (3%) What kind of algorithm is used to convert a regular tree to binary tree?

(8%) [Array]

4. (a) (5%) Please fill the blank to finish the matrix multiplication algorithm.

```
Matrix Multiply(Matrix a, Matrix b){
    Matrix M(a.rows, b.cols);
    for (i=0; i<a.rows; i++) {
        for (j=0; j<b.cols; j++) {
            sum=0;
            for (k=0; k<a.cols; k++)
                sum += ___(a)___;
            M[i][j]=sum;
        }
    }
    return M;
}
```

(b) (3%) Please calculate the time complexity of the above program using big-O.

(15%) [Stack]

5. (7%) Please convert the following infix expression to postfix expression using a **stack (Show detailed steps).**

$$A + B * (C - (D - (E - F)))$$

6. (8%) Please evaluate the following expressions using a **stack (**Write down the **type** of notation (prefix, infix, or postfix) and computed **result).**

For example: 1 2 3 4 * + + Answer: postfix, 15

(a) (4%) 9 7 * 2 5 * - 5 7 * -

(b) (4%) - * 3 - * 2 9 * 2 3 7

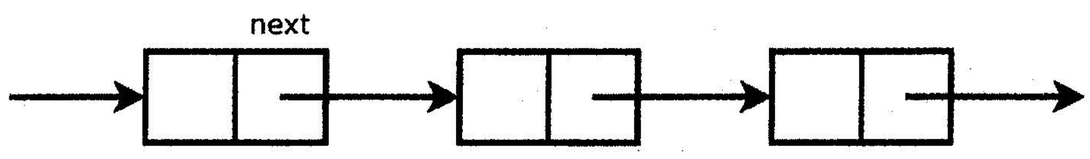
(12%) [Linked List]

Please answer the questions based on the following data structures:

```
//Singly-linked list node
class Node
{
public:
    int data;
    Node* next;
}
```

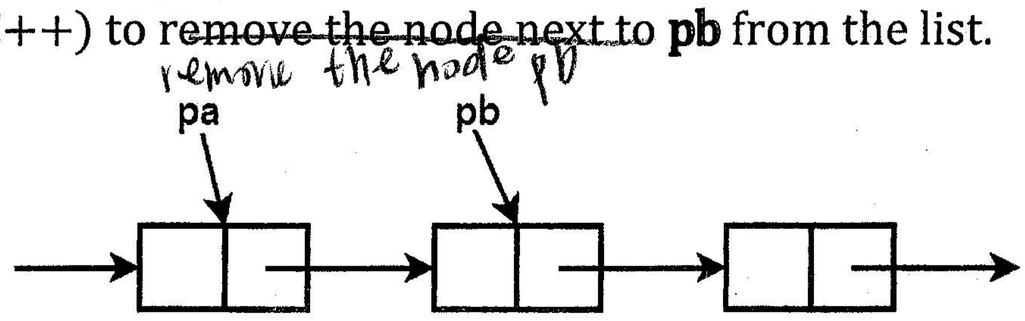
```
//Doubly-linked list node
class Node
{
public:
    int data;
    Node* prev;
    Node* next;
}
```

A singly-linked list is illustrated as below:

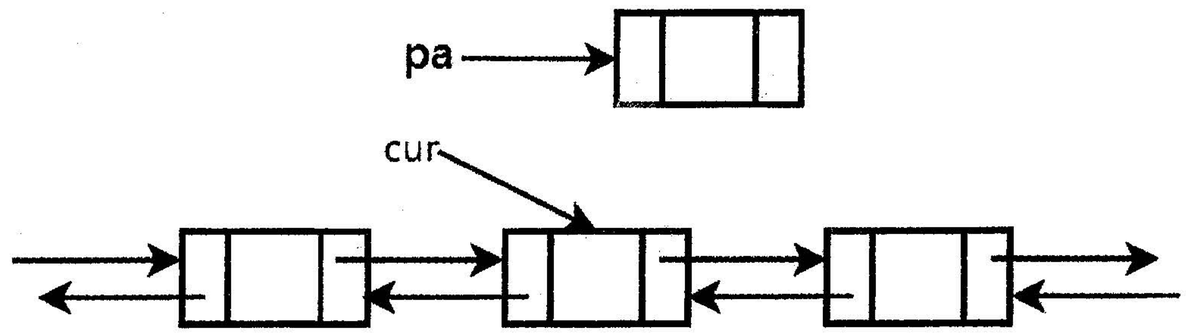


A doubly-linked list is illustrated as below:

7. (4%) The following diagram shows parts of a **singly-linked list** and its two nodes **pa** and **pb**. Please write pseudo codes(C/C++) to remove the node next to **pb** from the list.



8. (8%) The following diagram shows parts of a **doubly-linked list**. Please write pseudo codes(C/C++) to insert a new node **pa** into the list right after the node **cur**.

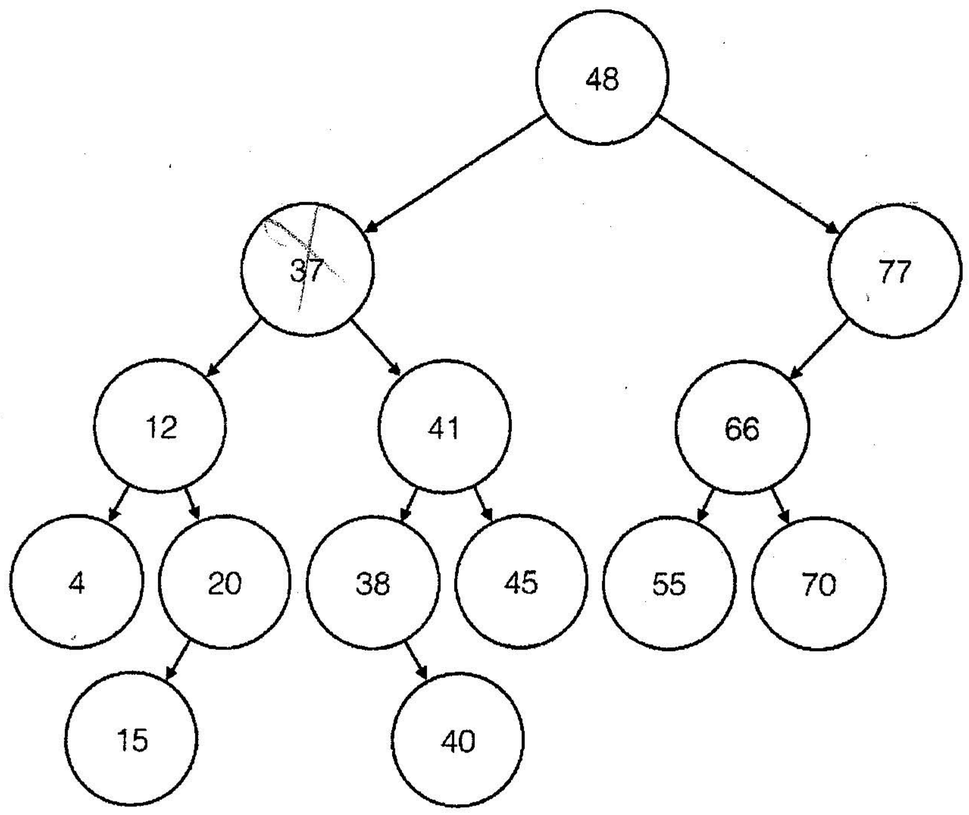


(50%) [Tree]

9. (6%) Please draw the original binary tree given the outputs of tree traversal.

(a) (3%)	In-order Traversal: AGCJBHI Pre-order Traversal: JGACHBI
(b) (3%)	In-order Traversal: GAHCIBJ Post-order Traversal: GHAIJBC

10. (10%) Given the following binary search tree (BST):



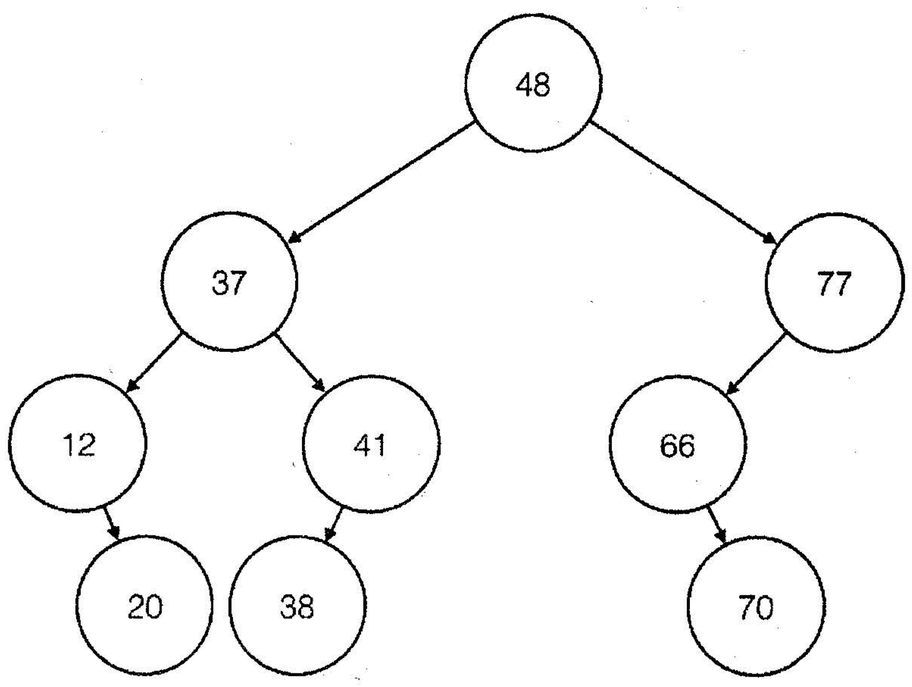
(a) (5%) Please draw the BST after deleting the node with key 37.

(b) (5%) Please draw the BST after inserting a node with key 60.

11. (8%) An **indexed binary search tree** is a BST with each node containing an additional data field **leftSize**, which is one plus the number of nodes in the left sub-tree, and the **rank** of a node is its position in the in-order traversal.

(a) (4%) Please explicitly label the **leftSize** and the **rank** of each node in the tree below.

(b) (4%) Draw the process of deleting the **fourth** smallest element utilizing **leftSize**.



12. (13%)

```
int Heap[1001];
```

A MAX heap is represented by an array, where the index starts from 1. Please write pseudo codes (C/C++) of the following operations.

(Note that the element number of MAX heap will not exceed 1000 or less than 1)

(a) (1%) findMAX-- return the max value of heap.

(b) (6%) deleteMAX -- delete the max value of heap.

(c) (6%) insert -- insert new value to heap.

(a) (1%)	(b) (6%)	(c) (6%)

int findMax(int Heap[])
{
//your code here
return max;
}
	void deleteMAX (int Heap[])
	{
	//your code here
	}
		void insert (int Heap[], int value)
		{
		//your code here
		}

13. (3%) Write pseudo codes to output the post-order traversal of a binary tree using the following data structure.

```
class Node{
public:
    int data;
      Node *lchild;
      Node *rchild;
}
```

14. (10%) A clocked tree is a Binary tree in which each node *ni* is associated with a non-negative delay, delay(*ni*). The path delay from a root to a node is defined as the summation of delay of all nodes along the path. The ***longestDelay*** is defined as the longest path delay among all root-to-leaf (terminal nodes) paths. Please write down the recursive procedure using pseudo codes (C/C++):
longestDelay(treenode *root, int AccumulatedDelay)
to compute the longest path delay **MAX**. Let the tree nodes be defined as:

```
class treenode{
int delay;
treenode *lchild;
treenode *rchild;
}
int MAX;
```