

Data Structures
Spring 2005, Midterm Exam II, May 3
7:00-8:50 PM

1. (10%) Let p be a pointer to the first node in a singly linked list. Write a procedure to reverse the order of nodes in the list. Do not create any additional nodes. Following the reversal, the original list should not exist. The following type definition is for the nodes in the list.

```
typedef struct listnode *listpointer;
typedef struct listnode {
    integer data;
    listpointer link;
};
```

2. (10%) Draw the Max-heap after the following operations are performed beginning with an empty heap: insert 43, insert 31, insert 68, insert 24, delete-max, insert 51, insert 44, insert 53, insert 69, insert 71, delete-max.
3. (10%) For an n nodes binary tree, what are its minimum and maximum heights?
4. (10%) Write a RECURSIVE C (or C-like) procedure *int count (treepointer t)* to compute the number of non-terminal nodes with only one child in a tree. The root of the tree is passed to the function through the parameter t . Let the tree node be defined as :

```
typedef struct treenode *treepointer;
typedef struct treenode {
    int key;
    treepointer lchild, rchild;
};
```

5. (10%) Let nodes of a Binary Search Tree be defined as

```
typedef struct treenode *treepointer;
typedef struct treenode {
    int key;
    int leftsize;
    treepointer lchild, rchild;
};
```

where leftsize is one plus the number of elements in the left subtree. Write a procedure *treepointer search (treepointer t, int k, int n)* which search the k th largest element, where n is the number of nodes in the tree pointed by t .

6. (10%) Let T be a rooted Binary tree and two nodes m and n in T . We can test whether a node m is a proper ancestor of a node n by testing whether m precedes n in X-order but follows n in Y-order, where X and Y are chosen from pre, post, in. Determine all those pairs X and Y for which this statement holds.
7. (10%) Give an n node Binary Search Tree structure on which both the find-min and find-max operations can be executed in constant time. (Clearly identify the smallest and the largest elements in your tree).

8. (10%)

- (a) Represent the following tree by a Binary tree representation.
- (b) If we store the above binary tree by one-dimensional array $A[1..n]$, what the indices of G , J are?

9. (10%) Assume that initially set i contains a single element i for $1 \leq i \leq 14$ and if $(|A| = |B|)$, then $A = A \cup B$. Draw the resultant tree after executing the following operations using `weightingUnion` and `CollapingFind`: `union(1, 2)`, `union(5, 7)`, `union(9, 11)`, `union(1, 3)`, `union(4, 5)` `union(6, 8)`, `union(10, 12)`, `union(9, 10)`, `union(13, 14)`, `union(6, 9)`, `union(1, 5)` `union(1, 13)`, `union(1, 9)`, `find(7)`, `find(12)`.

10. (10%) Write a RECURSIVE C (or C-like) procedure *int equal (treepointer first, treepointer second)* that takes two binary trees pointed by *first* and *second* and tests if two binary trees are equal. Let the tree node be defined as :

```
typedef struct treenode *treepointer;
typedef struct treenode {
    int key;
    treepointer lchild, rchild;
};
```