1. **a.** (10%) Write a pseudocode function that takes the head node of a linked list as input and moves the first largest item in the linked list to the end of the linked list. No return value is required.
   b. (10%) Write a pseudocode function that takes the head node of a linked list as input and returns the largest item in the linked list. Before returning, your function should remove all the nodes with that value from the linked list.
   Note: A **node** in a **linked list** has a pointer, named 'next', pointing to the next node.

2. (10%) Write a program (in pseudocode) that takes the root node of a **binary tree** as input and returns the number of leaves in the binary tree.

3. (15%) Write down the basic idea of re-constructing a **binary tree** by using its preorder and inorder traversals.

4. (15%) Describe the procedure of bottom-up **heap** construction. Prove that bottom-up heap construction takes linear time.

5. (10%) Write down the basic idea of implementing **priority queues** using a **heap**.

6. (15%) Suppose that you have a set of nodes with no **null** pointers (each node points to itself or to some other node in the set). Prove that you ultimately get into a cycle if you start at any given node and follow links.

7. (15%) Please briefly describe how to construct a **2d-tree** when given N 2-dimensional points. What are the **worst-case** and **typical-case** complexities of using a 2d-tree in a nearest neighbor search (find the closest point among N 2-dimensional points to the query point)?

試題隨卷繳回