Computer Science 221

Practice Questions - Set 3
Data Structures, etc.

1. Suppose that we use a linked list to represent a queue and that in addition to the enqueue() and dequeue() functions (i.e., functions to add and remove elements from the linked list), you want to add a new operation to the queue that **deletes the last element of the queue**. Which linked structure do we need to use to guarantee that this operation is also executed in constant time? Justify your answer.

2. Suppose the nodes of a binary tree structure are defined as:

```
struct Bnode
{
      int    value;
      Bnode* left;
      Bnode* right;
}
```

Define a function height that takes the address of a node in a binary tree and returns the height of the node in that tree. **[Recall:** The height of a node in a tree is the length of the longest path from the node to a leaf. The height of a node without children is 0, and for this function, if the user passes in a null node, you can assume that its height is 0, too.**]** Here's the function header to help you get started.

```
int height( Bnode* node)
{
```

3. What is the worst-case running time for inserting n items into an initially empty hash table, where collisions are resolved by chaining? What if each sequence is stored in sorted order?

4. Suppose that each row of an $n \times n$ array A consists of 1's and 0's such that, in any row of A, all the 1's come before any 0's in that row. Assuming A is already in memory, describe a function running in O(n log n) time (*not* $O(n^2)$ time) for counting the number of 1's in A.

5. Let A be a collection of objects. Describe (in words) an efficient function for converting A into a set. That is, remove all duplicates from A. What is the running time of this method?

6. Let G be a simple connected graph with n vertices and m edges. Explain why O(log m) is O(log n).