

# Computer Science 221

## Practice Questions - Set 3

### Solutions: Data Structures, etc.

1. Use a doubly linked list. Why? We need to update the node before the last.

2.

```
int height( Bnode* node)
{
    if ( node == NULL ||
        (node->left == NULL && node->right == NULL) )
        return 0;

    int leftHeight = height( node -> left );
    int rightHeight = height( node -> right );
    if (leftHeight > rightHeight)
        return leftHeight + 1;
    else
        return rightHeight + 1;
}
```

3. If the sequences are not sorted then the worst-case running time is  $O(n)$ . If they are stored in sorted order, then the worst-case running time is  $O(n^2)$ .
4. To count the number of 1's in A, we can do a binary search on each row of A to determine the position of the last 1 in that row. Then we can simply sum up these values to obtain the total number of 1's in A. This takes  $O(\lg n)$  time to find the last 1 in each row. Done for each of the  $n$  rows, then this takes  $O(n \lg n)$  time.
5. First, we sort the objects of A. Then, we can go through the sorted sequence and remove all duplicates. This takes  $O(n \lg n)$  time to sort and  $O(n)$  time to remove the duplicates. Overall, this is an  $O(n \lg n)$  algorithm.
6. We know that  $m \leq C(n,2) = n(n-1)/2$  is  $O(n^2)$ . It follows, therefore, that  $O(\lg(n^2)) = O(2 \lg(n)) = O(\lg n)$ .