



CSCI 591 Example Questions

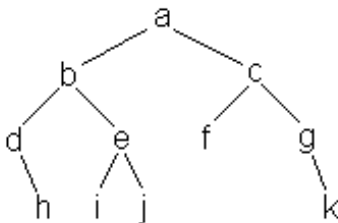


The following questions are representative of those that will appear on the CSCI 591 exam. This list is not exhaustive, and it is unlikely that any of these particular questions will be used.

- The harmonic series is $1 + (1/2) + (1/3) + (1/4) + \dots$. Write a recursive function that returns the sum of the first n terms of this series.
- Write a C++ function `insert_at(p, pos, value)` that inserts `value` at position `pos` in the list to which the pointer `p` points. Assume that the list initially contains at least `pos - 1` values, and use these declarations:

```
struct Node
{
    int item;
    Node *next;
};
```

- Given a list of integers implemented as a linked list as in the previous question, write a recursive function `sum(ls)` that returns the sum of the list to which the pointer `ls` points.
- Describe a linked implementation of a priority queue abstract data type in which the `dequeue()` operation is as fast as possible. We don't care about the times of other operations. What is the time complexity of the `dequeue()` operation in your implementation?
- Describe a stack-based algorithm for evaluating postfix arithmetic expressions and illustrate how it works by applying it to this expression:
`7 5 + 2 6 3 - * /`
- A palindrome is a string in which the letters read the same in both directions, disregarding capitalization. Describe an algorithm that uses a stack and a queue to determine if an input string is a palindrome.
- Describe how Merge Sort works and explain why its time is $O(n \log n)$.
- List the items in this binary tree in the order of a preorder traversal.



- Draw the binary search tree produced by inserting the following values into an initially empty tree in their listed order:
`58 23 16 72 32 37 42 64 51 25 90 68 53`
- The height of a tree is the number of edges in the longest path from the tree's root to any leaf. Write a recursive function `height(t)` that returns the height of the binary tree to which `t` points. Use these declarations:

```
struct Node
{
    int data;
    Node *left;
```

```
Node *right;  
};
```

- Sketch the weighted graph that this adjacency matrix represents:

0	5	0	3	6	0
5	0	0	0	7	0
10	0	0	0	9	6
3	0	0	0	0	15
6	7	9	0	0	4
0	1	6	16	4	0

- Starting at v_0 in the graph above, follow the strategy of Prim's algorithm to identify a minimum spanning tree on the graph.
 - How should a graph be represented so that a depth-first traversal on it is as fast as possible? Explain.
-