# CS 211 Lab Final Questions

Below is a list of potential questions that will be asked on the lab final. Note that one question on the lab final is not on the list below.

## Binary Nodes

Applicable questions will use the following definition for a binary node

|  |
| --- |
| class BinaryNode {  public:  int value;  BinaryNode \*left = nullptr;  BinaryNode \*right = nullptr;  BinaryNode(int v = 0) {  value = v;  }  }; |

## Linked Lists

Applicable questions will use the following definition for a linked list node

|  |
| --- |
| class LinkedListNode  {  public:  int \_value;  LinkedListNode \*\_next = nullptr;  LinkedListNode(int v = 0)  {  \_value = v;  }  }; |

1. Write a function that adds a given value into its proper place within a Binary Search Tree
2. Write a function that removes a given value from a given BST while maintaining proper BST structure
3. Write a function that determines whether or not a supplied vector is a binary heap
4. Write a function that merges two priority queues into a single sorted vector
5. Write a function that sorts an array using insertion sort, bubble sort, or selection sort (BONUS if you can instead write quick sort or merge sort)
6. Write a function that determines whether or not a given tree fulfills the BST rule
7. Write a function that determines whether or not a given tree is AVL compliant
8. Implement a function that parses the supplied vector of integers representing rainfall data and returns the day(as a string) that has the most rain. The input vector is formatted such that index 0 represents Sunday and every 7th item is the same day. Thus:
   1. 0, 7, 14… is a Sunday
   2. 1, 8, 15… is a Monday
   3. 2, 9, 16… is a Tuesday
   4. Etc.
   5. Example: given the array [1, 1, 2, 1, 1, 1, 3, 0, 0, 1, 4, 2, 1, 2] the day with the most rain is Saturday.
9. Write a function that converts a supplied BST into a sorted vector
10. Write a function that reverses a supplied linked list
11. Complete the function fizzbuzz() that returns a vector of size N (supplied input parameter) having values from 0 to N. However, when the index is divisible by 3, instead write "fizz". When the index is divisible by 5, instead write "buzz". When the number is divisible by both 3 and 5, write "fizzbuzz" to that index. For example, when N = 16, your vector should contain: 0, 1, 2, fizz, 4, buzz, fizz, 7, 8, fizz, buzz, 11, fizz, 13, 14, fizzbuzz
12. Write a function that finds the most commonly occurring string of length N in string S (similar to prior lab assignment)
13. Write a function that performs a right AVL rotation at the given node
14. Write a function that checks for balanced parenthesis. This time around, what constitutes a parenthesis will be supplied as an unordered\_map in the format var[open] = closed. E.g.
    1. Data['('] = ')'
    2. Thus, depending on what is supplied in this map will change the outcome of your program.