Final Exam

Your Name:

Instructions

Solve each of the following problems to the best of your abilities. You should write your answers on the scratch sheets of paper provided and then staple them to your exam booklet. Show all your work for full credit. Once you have completed the exam, hand it in to me, and you can leave for the day.

The exam is worth 125 points and is calibrated for 100 minutes.

Good luck!

1. Write the pseudocode for an algorithm that takes the root of a binary search tree as its only argument and returns the sum of all of the values of the nodes of the tree. You can assume that the tree stores integers as its node values.

2. What does it mean when I say that an algorithm is “in place”?

3. Describe the strategy used by a splay tree to rearrange nodes after an insertion / search. I do not need to see any math or code – just a high-level explanation is sufficient.

4. Suppose I wanted to list the values stored in a binary search tree in ascending order. Would I use an in-order, pre-order, or post-order traversal?

5. Suppose I have a binary tree data structure which is NOT a binary search tree. I write an algorithm to search for the largest element in the tree by looking at the left and right children of a given node and recursively visiting the largest child until I reach a leaf. Is this an example of a brute force, greedy, or dynamic programming algorithm? Why?

6. What is the difference between a multiset and a set data structure?

7. Oftentimes when we use the quicksort algorithm, we will introduce randomization to the list during sorting. We might pick a random element of the input sequence as the pivot each time we partition, or we might randomize the list beforehand. Why do we introduce randomization when we run the quicksort algorithm?

8. A hacker has stolen a table of hashed passwords and wants to reverse engineer user’s data. For a given hash value, they try every possible combination of characters to match the password with the hash (e.g. a, aa, ab, ac, …). Is this an example of a brute force, greedy, or dynamic programming algorithm? Why?

9. What is the difference between a divide and conquer versus a decrease and conquer algorithm?

10. Set A contains elements {1, 4, 5, 6, 7}. Set B contains elements {1, 2, 3, 4, 5}. What is the union of set A and set B?

11. Suppose I create a probabilistic skip list data structure. After insertion of an element, the data structure "flips a fair coin" repeatedly until the first instance of tails appears. The height of the inserted element in the skip list is equal to the number of heads counted before tails appears. How would you go about calculating the expected height of the element after the insertion? I do not need to see any math or code – just a high-level explanation is sufficient.

12. What is a Huffman code? What is it used for? I do not need to see any math or code – just a high-level explanation is sufficient.

13. Prove that the time complexity of the selection sort algorithm is $O(n^2)$ using its recurrence relation. The selection sort algorithm has a recurrence relation given below, where c_1 and c_2 are constants:

$$T(n) = T(n-1) + c_1 n$$

$$T(0) = c_2$$

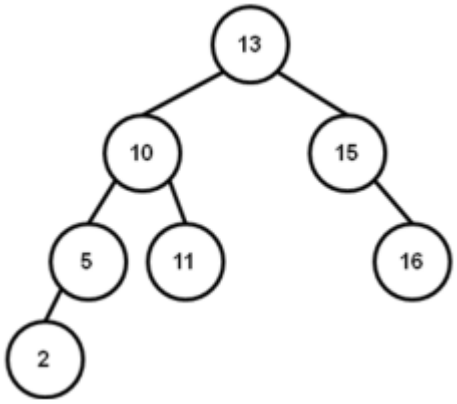
I want to design a compression algorithm for a hashmap using the division method. Recall that this method maps an integer x to a range from [0, N] according to the following function:

$$f(x) = x \bmod N$$

The size of the bucket array for my hashmap is N = 7, and it uses separate chaining to handle collisions.

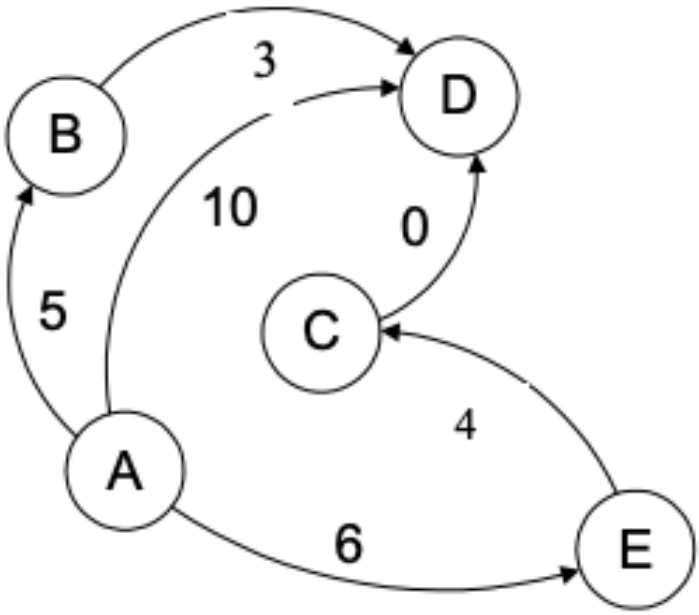
1. Sketch out the final state of the bucket array after inserting the numbers 4, 8, 2, 9, and 11.
2. Why is it important that the data stored in a bucket array be as evenly distributed over the buckets as possible?
3. Suppose I inserted 0, 7, 14, 21, 28, etc. into the hashmap described above. How would this affect the overall performance of the hashmap? Why?

Consider the AVL tree shown below.



1. What is the balance factor of node 15?
2. What is the height of node 10?
3. Starting from the original tree shown above, sketch the final state of the tree after the number 4 is inserted and the tree rebalanced.
4. Starting from the original tree shown above, sketch the final state of the tree after the number 11 is deleted and the tree rebalanced.

Consider the weighted graph shown below.



1. Is this a directed or undirected graph?
2. Is node B reachable from node E?
3. What is the shortest path from node A to node D?
4. Write out the edge list for the graph.
5. Write out the adjacency matrix for the graph.

Extra Credit Questions

1. I want to use a 4-bit cyclic shift to compute a hash code for a number. What is the hash code for the number 2000? Recall that 2000 is 11111010000 in binary.

2. Sketch out the trie that stores the words “badger”, “badge”, “barge”, and “computer”.

3. How does the Boyer-Moore algorithm improve upon the naive $O(n^2)$ performance of pattern matching via brute force?

4. What problem does the Randomized Quick Select algorithm solve? How does it work? I do not need to see any math or code – just a high-level explanation is sufficient.

5. Consider the red-black tree below. Sketch out the new tree that would result if I inserted the number 7.