Midterm 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 100 points and is calibrated for 75 minutes.

Good luck!

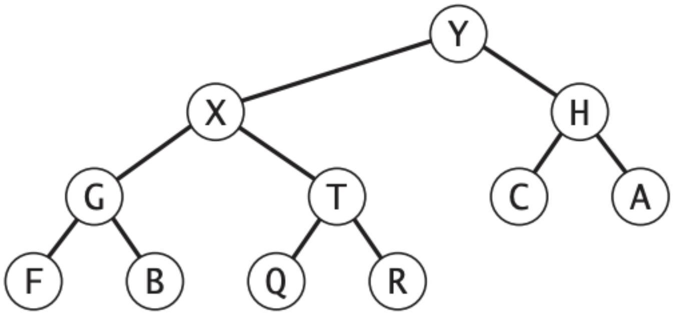
1. Evaluate the following postfix expression: 3 1 - 4 2 + *
2. What would happen if you ran Floyd’s cycle finding algorithm on a singly linked list that does NOT have a cycle in it?
3. What is the difference between a FIFO and a LIFO data structure? Give an example of each.
4. A complete binary tree contains 13 nodes. What is the height of the tree? Show your work and / or sketch a diagram.
5. A perfect binary tree has seven internal nodes. How many external nodes does it have? Show your work and / or sketch a diagram.
6. Prove that f(n) = 5n + 4 is O(n). Show your work.
7. Suppose I create a two-dimensional array with three rows and six columns. What is the index of the element in row r = 2 and column c = 3 if the language uses column-major ordering?
8. What is the transitive property in the context of ordering elements in a priority queue?
9. What is a “black hole” in the context of benchmarking code? Why is it important?
10. Prove the following statement using induction:

$$\sum_{i=0}^{n}(3+5i)=\frac{(n+1)(5n+6)}{2}$$

Suppose you have a doubly linked list of unknown size that is populated with integers. You have access to the HEAD and TAIL pointers of the list, as well as operations to access and traverse nodes in the list.

1. Write the pseudocode for an algorithm that adds up all the numbers in the list and returns the final sum. Be sure to handle the edge case when the list is empty.
2. What is the time complexity of your algorithm? Prove your answer.
3. How would your algorithm change if the doubly linked list used sentinel nodes to mark its boundaries? You don't have to re-write the pseudocode, you can just describe the changes with a sentence or two.

Consider the max heap shown in the diagram below. Letters are ordered by their ASCII codes – i.e. letters that come later in the alphabet are “larger” than letters that come earlier in the alphabet.



1. Write out the array representation of the heap.
2. What would I get if I printed out the elements of the binary tree using a pre-order, depth first traversal?
3. Suppose I insert the letter “S” into the heap. Sketch the tree diagram for the new heap after the heapify operation is complete.

Consider the pseudocode below which describes a recursive function.

```
int my_function(int a, int b) {
 if (a % b == 0) {
  return b;
 }
 return my_function(b, a % b);
}
```

1. Sketch a recursive trace of my_function(17, 3).
2. Is this an example of linear recursion, binary recursion, or multiple recursion? How do you know?
3. Is this a tail-call optimized recursive function? Why or why not?