CS 380 Final Exam Review Questions

Explain what g(x) = BIG THETA(f(x)) means.

Explain what g(x) = BIG OMEGA(f(x)) means.

Explain what g(x) = BIG OH(f(x)) means.

$2n + 10000 = O(n)$: True or False? Justify your answer.

$2n + 10000 = O(n^2)$: True or False? Justify your answer.

Why do computer scientists like runtimes such as O(lg n)?

Is insertion sort an incremental algorithm? What properties does an incremental algorithm have?

Needleman-Wunch is an incremental algorithm. True or False. Justify your answer.

You have a set of input that you need to sort. The input is at least 1 million items and is either almost correctly sorted or sorted in reverse order. Which sort algorithm would you use and why?

What is the recurrence that represents the runtime of merge sort? For each term in the recurrence, explain how that term is reflected in the MergeSort pseudo code.

Solve the recurrence: $T(n) = T(n/2) + 1$; $T(1) = 1$
Use iteration and then induction.

If we know that algorithm X has a runtime of $O(n^3)$ in the worst case, what do we know about the best case runtime of X? Circle all that are correct and justify each answer.

a. The best case runtime is $O(n^3)$

b. The best case runtime is $\mathbf{\Omega}(n^3)$

c. The best case runtime is $\mathbf{\Omega}(n^2)$

d. We know nothing about the best case runtime

e. None of the above

2. In a full, complete binary tree where all leaves are at the same level and there are *n* nodes:

   a. How many nodes have zero children?
   b. How long is the longest path from the root to the deepest leaf?

What is the heap property for min-heaps?
If a heap is sorted in an array, what array index contains the parent of the element at array index i?

Why is it easy to store a heap as an array but difficult to store a red black tree as an array in the same manner?

What is the runtime of heap-maximum()?

What is the runtime of heap-extract-maximum()?

What is polymorphism?

How does the strategy pattern take advantage of polymorphism?

Use an iterator to print every int in this vector:
std::vector<int> theVector;

Use an iterator to print every int in this vector, in the opposite order from the previous solution:
std::vector<int> theVector;

Use an iterator to increment every int in this vector:
std::vector<int> theVector;

Why is the partition function so important for quicksort? Explain the partition function's impact on the runtime of quicksort.

Is quicksort an incremental algorithm? If so, explain the algorithm's incremental nature. If not, explain why not.

What is the runtime of counting sort? Is this a stable, in-place sorting algorithm?

Why does radix sort rely on a stable sort to be used for each digit?

How many comparisons are needed to find the smallest element out of 10 elements?

How is randomized-select related to quicksort?

Explain, in full English sentences, how randomized-select works.

How are comparison based and non-comparison based sort algorithms different? Give an example of each type of algorithm.

What are the properties of a Red-Black tree?

Insert the following values in to a Red-Black tree (drawing a tree after each insertion): 1 2 3 4 9 8 7 6

In RBDelete, y-original-color is checked and RB-Delete-Fixup is called only if y-original-color is Black. In what instances would y-original-color be Black? Why would we need to call the fixup method in each of those instances?

What problem characteristics might lead you to use an adj matrix? and adj list?

How are BFS and Dijkstra's algorithm similar?

What is the running time of BFS?

What is the purpose of the Queue in BFS? How would the algorithm change if the Queue was replaced by a stack?

Draw a graph and a MST for that graph where the MST contains a path between two nodes that is not the shortest path between those two nodes.

What are the properties of an MST?

Explain what a cut is in a graph. What is a light edge?

How many edges are there in an MST? How do we know that?

What is a disjoint-set data structure? Why is this important to Kruskal?

When applying Prim's algorithm, does it matter which node we pick to start with? Can we pick a node that does now allow us to form an MST using Prim's algorithm?

How does Horspool's search algorithm speed up searching?
What are the four cases in Horspool's algorithm used to determine how to shift the pattern?
Build the shift table for the pattern:
"computerscience"

Dijkstra's and Prim's algorithms both fill a Priority Queue (min-queue) with nodes and slowly extract the nodes from the queue. What does the key for the nodes represent in each algorithm? Do both algorithms produce the same set predecessors for each node? Why or why not?

Provide the four steps for Dynamic Programming for the Rod Cutting algorithm.

Provide the four steps for Dynamic Programming for the Longest Common Subsequence algorithm.

What does the value located at c[i,j] represent in the definition of the Longest Common Subsequence algorithm?

What does the value located at b[i,j] represent in the definition of the Longest Common Subsequence algorithm?

The Levenshtein Edit Distance definition is very similar to the LCS definition, however, there is a min() surrounding the (lev a,b (i-1,j)+1, lev a,b (i,j-1)+1, lev a,b (i-1,j-1)+ ai != bj). What does this min() achieve? How might this min() cause the Edit distance calculation to differ from the LCS calculation for the same pair of strings?

What is the definition of a matching?

What is the definition of an alignment?

Be sure you are able to explain/discuss your solutions to the questions in the final project (DNA & strings).

Be sure you are able to explain/discuss your design for the Path Finder project.

Write two questions, and their answers, that you think could legitimately be on the final exam. Post those questions to Moodle.