CS380: Algorithm Design and Analysis
Review for Final Exam

Exam Monday, May 15th, 2017 from 3:00-5:30pm
Material Covered: Lectures 1-21

In addition to what is listed below, study the previous three exam reviews in addition to the previous three exams and two quizzes. Reviews are on the course website, solutions to exams and quizzes on Grace in CS380 Public.

Topics:
Asymptotic Analysis
Insertion sort
Merge Sort, Recurrence Equations, Divide and Conquer versus Incremental Algorithms
Solving Recurrences, Master Method
Quicksort (various pivots, median of three)
Heapsort (inc. Binary Trees)
Priority Queues
Linear Sorting (Comparison, Counting, Radix, Bucket)
Order Statistics
Red-Black Trees (Insert, Fixup)
Augmenting Data Structures (example O(lg n) runtime for finding order statistics)
Dynamic Programming and Rod Cutting
LCS
Edit Distance
Sequence Alignment (Needleman-Wunsch)
Horspool
BFS, DFS on Graph
Single Source Shortest Path (Bellman Ford, Dijkstra)
Minimal Spanning Trees (Kruskal, Prim)
Topological Sort (as example of use of DFS)

Note: Most algorithm’s will be provided on the exam, except in situations where would trivialize the question.

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(n²): 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?

Insertion sort is a $O(n^2)$ algorithm. Under what conditions is it preferable to Merge Sort, which is a $O(n \ast \lg(n))$ algorithm? Why?

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 Merge Sort pseudo code.

Solve the recurrence: $T(n) = T(n/2) + 1$ ; $T(1) = 1$

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

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 $i$? Similarly, what array indices contain the left and right children, respectively, of the parent node whose index is $j$?

What is the runtime of Max-Heapify? (recursion results from the most unbalanced complete binary tree possible).

What is the runtime of Build-Max-Heap? Of HeapSort, assuming Build-Max-Heap has already been run?

What is the runtime of Heap-Maximum ()?

What is the runtime of HEAP-EXTRACT-MAXIMUM()? (including call to Max-Heapify).

What is the role of Heap-Increase-Key in the Max-Heap-Insert algorithm?

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

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

What is the ith order statistic of a list of n elements? What is the worst-case runtime of the naïve algorithm for finding the ith Order Statistic?

How is randomized-select related to quicksort?

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

What does the SELECT algorithm accomplish that the randomized-select algorithm does not?

What are the properties of a Red-Black tree?

What is the significance of the fact that a red-black tree with n internals nodes can have a height no greater than \(2\log(n+1)\)?

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

What is the significance of the runtime of the OS-SELECT(x,i) algorithm?

What issue does dynamic programming attempt to solve?

Is dynamic programming always used with any recursive algorithm? If not, what is an example of when it is not used?

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

What does memorization mean?

When should we use Dynamic Programming instead of a Divide and Conquer 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 \((d(i-1,j)+1, d(i,j-1)+1, d(i-1,j-1)+1)\) expression when \(x_i \neq y_j\). 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?

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: “summerfinally”

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?

What problem characteristics might lead you to use an adj matrix? an 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 were 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 the difference between Prim’s Algorithm and Kruskal’s algorithm with regard to how they build the MST?