COS 226 Design Exam Fall 2016

There are four questions on this exam, weighted as indicated at the bottom of this page.

Policies. The exam is closed book, though that you are allowed to use a single-page one-sided hand-written cheatsheet. No calculators or other electronic devices are permitted. Give your answers and show your work in the space provided. You will have 80 minutes to complete the test. **This exam is preprocessed by computer.** Write all answers inside the designated rectangles. Do not write on the corner marks.

This page. *Print your name, login ID, and precept number on this page* (now), and write out and sign the Honor Code pledge.

Discussing this exam. As you know, discussing the contents of this exam before solutions have been posted is a serious violation of the Honor Code.

Name

Login

Precept

"I pledge my honor that I have not violated the Honor Code during this examination."

[*copy the pledge here*]

[*signature*]

Q1	Q2	Q3	Q3	TOTAL
/8	/8	/8	/6	/30

Q1. Prefix-free codes. In data compression, a set of binary code words is *prefix-free* if no code word is a prefix of another. For example, the set of codewords `{01, 10, 0010, 1111}` is prefix-free, but `{01, 10, 0010, 10100}` is not because `10` is a prefix of `10100`.

A. Design an efficient algorithm to determine if a set of binary code words is prefix-free. *You do not need to provide code, just describe how you would use the algorithms and data structures you have studied, with suitable modifications.* Your answer will be graded on correctness, efficiency, clarity, and succinctness.

B. What is the order of growth of the worst-case running time of your algorithm as a function of N and W, where N is the number of binary code words and W is the total number of bits in the input?

N	W	NW	$NW \log N$	$N \log N$	N^2
○	○	○	○	○	○

C. What is the order of growth of the memory usage of your algorithm?

N	W	NW	$NW \log N$	$N \log N$	N^2
○	○	○	○	○	○

Q2. 1D nearest neighbor. Consider the following API for the *1D nearest neighbor* problem.

constructor	create a data structure containing the real number 0.0
`insert(x)`	insert the real number `x` into the data structure
`query(y)`	return the real number in the data structure that is closest to `y`

Describe how you would build a data type that implements this API. For simplicity, assume that the values processed are all different. For full credit, all operations should take time logarithmic in the number of insertions in the worst case. *You do not need to provide code, just describe how you would use the algorithms and data structures you have studied, with suitable modifications.* Your answer will be graded on correctness, efficiency, clarity, and succinctness.

A. constructor

B. `insert(x)`

C. `query(y)`

Q3. Shortest path with landmark.

A. Design a data type that preprocesses a given directed graph G with positive edge weights, a given source vertex v, and a given landmark vertex x so as to be able to efficiently respond to client queries that give a destination vertex w and ask for the length of the shortest path from v through x to w. (For example, FedEx might use this with x equal to Atlanta.) For full credit, you should use space linear in the number of vertices in G and answer queries in constant time. *You do not need to provide code, just describe how you would use the algorithms and data structures you have studied, with suitable modifications.* Your answer will be graded on correctness, efficiency, clarity, and succinctness.

B. Design another data type that takes only G and x for preprocessing and can efficiently answer queries that give both v and w and ask for the length of the shortest path from v to w through x. For full credit, you should use space linear in the number of vertices in G and answer queries in constant time.

Q4. MST with weight restrictions.

A. Suppose that you need to process a weighted undirected graph with V vertices and $50V$ edges, where the edges weights all are one of three positive `double` values x, y, or z. Describe an efficient algorithm for finding an MST of such a graph. For full credit, you must give the the worst-case running time of your algorithm and it must be asymptotically better than $V \log V$.

B. Answer the same question with the assumption that edge weights are positive `int` values.