CPSC 221: Algorithms and Data Structures
Assignment #3, due never! For practice only...

Submission Instructions

Do not submit this "assignment"! This is just for you to get some practice on some of the material from the end of the course.

We suggest that you try to do these on your own, first, as best as you can. Then, we'll have "SPOILER" threads on Piazza, for you all to discuss and compare your answers and converge on the correct solutions (which the profs will bless when you've converged).

Questions

[11] 1. In this problem, you will draw Dictionary data structures *after* each "interesting" insertion is complete from the following list of insertions: 21, 31, 41, 51, 61, 71, 81, 91, 101, 111, 121, 131, 104, 108, 106. An "interesting" insertion is defined for each part.

(a) A Binary Search Tree (with no self-balancing). Only the last insertion is interesting.

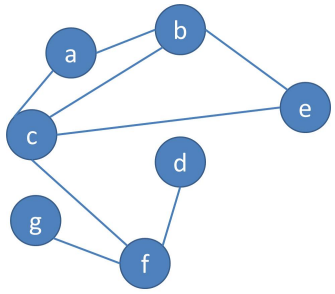
(b) An AVL Tree. Every insertion that causes a rebalancing is interesting, as is the last. (This one will require more diagrams than the others, but there should still be no more than 10 interesting insertions.)

(c) A Hash Table of size 11 that uses chaining (with unsorted linked-list chains, recently inserted items inserted at the front of the chain). Hash values by modding by the table size. Every insertion that collides is interesting, as is the last insertion.

(d) A Hash Table of size 23 that uses open addressing with double hashing. The first hash is modding by the table size. The second hash is $h_2(n) = 13 - (n \mod 13)$. Every insertion that collides is interesting, as is the last insertion.

[13] 2. Consider the following three graphs presented in three different forms and answer each of the questions below about them.

GRAPH A:



GRAPH B:

Bettina	Nancy	Paris	Renee	
Dan	Li			
Ganesh	Li			
Li	Dan	Ganesh	Paris	
Nancy	Bettina	Paris		
Paris	Bettina	Li	Nancy	Renee
Renee	Bettina	Paris		

GRAPH C:

	1	2	3	4
1	0	1	1	0
2	1	0	0	1
3	1	0	0	1
4	0	1	1	0

(a) Draw graphs B and C. (I.e., make pictures for them somewhat like Graph A's picture.)

(b) Convert graphs A and C into adjacency lists (like graph B).

(c) Convert graphs A and B into adjacency matrices (like graph C).

(d) Which pairs of these graphs are isomorphic? Indicate how their nodes line up.

(e) Which of these graphs is a subgraph of one (or more) of the others? Indicate which nodes/edges of the supergraph are represented in the subgraph. (WARNING: this is, in part, a trick question.)

(f) Draw a spanning tree for A.