CSE 332: Data Abstractions
Assignment #5
November 8, 2010
due: Monday, November 15, 2:30 p.m.

1. The `remove` procedure of Figure 5.17 for deletion from a hash table with open addressing uses “lazy deletion”.
   (a) Starting from an empty hash table, give an example with *as few dictionary operations as possible* that demonstrates that using “full deletion” can cause the hash table to return the incorrect result for some operation. Make your example complete:
      - State the table size, probing strategy, and hash function.
      - Provide the sequence of operations and the state of the hash table after each operation.
      - Demonstrate how lazy deletion leads to the correct result.
      - State the incorrect result that will occur using full deletion.

   (b) When rehashing to a larger table, do lazily-deleted items need to be included? Explain your answer.
2. Exercise 6.2. Show your heaps using trees rather than arrays. Show the heap after every insertion in part (a) and after every `percolateDown` that changes the heap in part (b).
3. Exercise 6.3. Just apply these operations starting with the tree from Exercise 6.2(b). Show the tree after each `deleteMin`.
4. Give an algorithm (in pseudocode or Java) that outputs all keys less than x in a binary heap, without changing the heap. The keys need not be output in sorted order. Your algorithm should run in time $O(L)$, where L is the number of keys that are output; note that this generalizes the fact that `findMin` runs in time $O(1)$. (Hint: recursion will help.)
5. (a) Given an acyclic directed graph $G = (V, E)$ representing course prerequisites, describe an algorithm that computes a schedule for completing all the courses in the minimum number of academic terms, with each course completed in the earliest possible term. Your algorithm should assign a term number `v.term` to every vertex `v`, beginning with term number 1. Assume that there is no limit on how many courses can be taken in any given term and that every course is offered every term.

   (b) What is the asymptotic running time of your algorithm in terms of $n = |V|$ and $e = |E|$? Justify your answer.
6. Exercise 9.5. Show your work for part (a) as in Figures 9.21-9.27 and for part (b) as in Figure 9.19.