CSE 332: Data Abstractions
Assignment #5
October 24, 2014
due: Monday, November 3, 12:30 p.m., *before lecture begins*

Bundles: The problems in each written homework assignment will be divided into 2 groups (to facilitate distribution to grading TAs). You will turn in 2 corresponding bundles. Write your full name in the *upper left corner* of each bundle's top page, with your last name printed clearly in CAPITAL LETTERS. Each bundle should be stapled separately. We don't supply the stapler.

This week's turnin bundles: (A) problems 1–2, (B) problems 3–4.

1. For this problem, you are using the hash function $h(x) = x \bmod 13$ with no rehashing. Insert the integers 87, 19, 25, 55, 36, 46, 88, 7, 67, 21 (in this order) into an initially empty hash table of size 13,
   (a) using separate chaining.
   (b) using open addressing with linear probing.
   (c) using open addressing with double hashing, where $h_2(x) = 1 + (x \bmod 12)$.
   (d) If a hash table is going to be this full (i.e., $n \approx m$) most of the time, and you have a hash function that spreads the entries over the hash table reasonably well, which of these methods is likely to be fastest, and why?

2. 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.

3. Solve the following recurrence for $T(n)$:

$$\begin{aligned} T(1) &= c \\ T(n) &= 2T(n/2) + d \end{aligned}$$

   where c and d are constants independent of n.

4. (a) Given an acyclic directed graph $G = (V, E)$ representing course prerequisites, write 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) Show the result of running your algorithm on the graph of Figure 9.3.

   (c) What is the asymptotic running time of your algorithm in terms of $n = |V|$ and $e = |E|$? Justify your answer.