CS 410 Summer 2000
Homework 4
Due 11:30 AM, Monday July 24

Important notes about all exercises for this course:

- No late homework will be accepted.
- Justify your answers. Answers without brief (but adequate) justification may be considered incomplete.
- You may use pseudocode, Java, C, or C++ to write your algorithms. You do not need to compile or run your code for written exercises.
- When presenting algorithms, also provide a brief English description of the pseudocode and an explanation of why the algorithm (and your analysis of it) is correct.

Reading

You should already have read CLR chapters 9, 11, 12.
Read CLR chapters 13.1, 13.2, 13.3, 14, 19

Written Exercises

1. **Problem 1 will be distributed tomorrow (Thursday, 7/20).**

2. The following figure depicts a linked list implemented with an array.

   list $\boxed{4}$ free $\boxed{3}$

Index	Key	Next
1	12	0
2	19	5
3	8	2
4	6	10
5	14	1
6	32	7
7	67	9
8	95	0
9	68	8
10	11	6

   (a) What elements are in the list?
   (b) What array positions are part of the free-space list?
   (c) What would the array look like after deletion of 68?
   (d) What would the array look like after insertion of 17?

3. Suppose you are given the following information about a hashtable.

Space Available (in words)	10000
Words per Item	7
Words per Pointer	1
Number of Items	1000
Proportion Successful Searches	1/3

Which hashing method, chaining or double hashing, looks best (i.e., the one that we expect to use the least amount of time)? You should use the following assumptions:

- It takes twice as long to compute a hash function as it does to do a key comparison.
- For hashing with chaining use singly linked list.
- Double hashing behaves like random hashing.
- During double hashing, we always compute both hash functions, even when we find the item we're looking for on the first probe.
- Items are not copied directly into the hashtable; instead we maintain a link to the item. Note though that the items themselves still use up some of the space.

As always, justify your answer.

4. How many 5-element binary search trees are there with the vowels "A", "E", "I", "O", and "U" as the keys (note ordering is by alphabetical order)? You not have to draw them (but as always you must justify your answer).

5. CLR 13.3-4.

6. Describe a general way to construct red-black trees that are as unbalanced as possible. By "most unbalanced," we mean that the height of a tree constructed this way should be as close as possible to the number of nodes in the tree. (You do not need to give pseudocode for your method, just describe your construction, and explain why it results in the most unbalanced red-black trees. You may use pictures, if you like.)

   Give a lower bound for h in terms of n (height of the tree and number of internal nodes, respectively) for your construction. Remember that in general the relation is $h \leq 2\log_2(n+1)$.

7. Draw all possible red-black trees containing keys 1, 2, and 3. Observe the convention that the root node should be colored black. How many different ways are there to do this?

8. CLR 14.2-3.

9. CLR 19.1-5.

10. CLR 19.2-1.

11. CLR 19.2 on page 399.