Problem Set 4

MIT students: This problem set is due in recitation on *Day 13*.

Reading: Chapters 10 and 11

Both exercises and problems should be solved, but *only the problems* should be turned in. Exercises are intended to help you master the course material. Even though you should not turn in the exercise solutions, you are responsible for material covered by the exercises.

Mark the top of each sheet with your name, the course number, the problem number, your recitation instructor and time, the date, and the names of any students with whom you collaborated.

MIT students: Each problem should be done on a separate sheet (or sheets) of three-hole punched paper.

You will often be called upon to "give an algorithm" to solve a certain problem. Your write-up should take the form of a short essay. A topic paragraph should summarize the problem you are solving and what your results are. The body of your essay should provide the following:

1. A description of the algorithm in English and, if helpful, pseudocode.
2. At least one worked example or diagram to show more precisely how your algorithm works.
3. A proof (or indication) of the correctness of the algorithm.
4. An analysis of the running time of the algorithm.

Remember, your goal is to communicate. Graders will be instructed to take off points for convoluted and obtuse descriptions.

Exercise 4-1. Do exercise 10.1-6 on page 204 of CLRS.

Exercise 4-2. Do exercise 10.2-4 on page 208 of CLRS.

Exercise 4-3. Do exercise 10.3-4 on page 213 of CLRS.

Exercise 4-4. Suppose we hash elements of a set U of keys into m slots. Show that if $|U| > (n-1)m$, there is a subset of U of size n consisting of keys that all hash to the same slot, so that the worst-case searching time for hashing with chaining is $\Theta(n)$.

Exercise 4-5. Do exercise 11.3-3 on page 236 of CLRS.

Problem 4-1. Comparisons among dynamic sets

For each type of dynamic set in the following table, what is the asymptotic running time for each operation listed, in terms of the number of elements n?

For operations that have not been explicitly defined, consider how you would implement the operation given the data structure. You do not need to give the algorithm, just the running time. State any assumptions that you make.

Assume that the hash tables resolve collisions by chaining with doubly linked lists.

	unsorted singly linked list, worst-case	sorted doubly linked list, worst-case	min-heap, worst-case	hash table, worst-case	hash table, average-case
SEARCH(L, k)					
INSERT(L, x)					
DELETE(L, x)					
SUCCESSOR(L, x)					
MINIMUM(L)					
MAXIMUM(L)					

Problem 4-2. k-universal hashing and authentication

Let $\mathcal{H}$ be a class of hash functions in which each hash function $h \in \mathcal{H}$ maps the universe U of keys to $\{0, 1, \ldots, m-1\}$. We say that $\mathcal{H}$ is k-***universal*** if, for every fixed sequence of k distinct keys $\left\langle x^{(1)}, x^{(2)}, \ldots, x^{(k)} \right\rangle$ and for any h chosen at random from $\mathcal{H}$, the sequence $\left\langle h(x^{(1)}), h(x^{(2)}), \ldots, h(x^{(k)}) \right\rangle$ is equally likely to be any of the m^k sequences of length k with elements drawn from $\{0, 1, \ldots, m-1\}$.

(a) Show that if the family $\mathcal{H}$ of hash functions is 2-universal, then it is universal.

(b) Suppose that the universe U is the set of n-tuples of values drawn from $\mathbb{Z}_p = \{0, 1, \ldots, p-1\}$, where p is prime. Consider an element $x = \langle x_0, x_1, \ldots, x_{n-1} \rangle \in U$. For any n-tuple $a = \langle a_0, a_1, \ldots, a_{n-1} \rangle \in U$, define the hash function h_a by

$$h_a(x) = \left(\sum_{j=0}^{n-1} a_j x_j \right) \bmod p \, .$$

Let $\mathcal{H} = \{h_a\}$. This is the family of hash functions shown in lecture to be universal. Show that $\mathcal{H}$ is not 2-universal. (*Hint:* Find a key for which all hash functions in $\mathcal{H}$ produce the same value.)

(c) Suppose that we modify $\mathcal{H}$ slightly from part (b): For any $a \in U$ and for any $b \in \mathbb{Z}_p$, define

$$h'_{a,b}(x) = \left(\sum_{j=0}^{n-1} a_j x_j + b \right) \bmod p \, .$$

and $\mathcal{H}' = \left\{ h'_{a,b} \right\}$. Argue that $\mathcal{H}'$ is 2-universal. (*Hint:* Consider fixed $x \in U$ and $y \in U$, with $x_i \neq y_i$ for some i. What happens to $h'_{a,b}(x)$ and $h'_{a,b}(y)$ as a_i and b range over $\mathbb{Z}_p$?)

(d) Suppose that Alice and Bob secretly agree on a hash function h from a 2-universal family $\mathcal{H}$ of hash functions. Each $h \in \mathcal{H}$ maps from a universe of keys U to $\mathbb{Z}_p$, where p is prime. Later, Alice sends a message m to Bob over the Internet, where $m \in U$. She authenticates this message to Bob by also sending an authentication tag $t = h(m)$, and Bob checks that the pair (m, t) he receives satisfies $t = h(m)$. Suppose that an adversary intercepts (m, t) en route and tries to fool Bob by replacing the pair with a different pair (m', t'). Argue that the probability that the adversary succeeds in fooling Bob into accepting (m', t') is at most $1/p$, no matter how much computing power the adversary has, even if the adversary knows the family $\mathcal{H}$ of hash functions used.