Problem Set 4

All parts are due on October 10, 2017 at 11:59PM. Please write your solutions in the LaTeX and Python templates provided. Aim for concise solutions; convoluted and obtuse descriptions might receive low marks, even when they are correct. Solutions should be submitted on the course website; there is no coding to submit for this problem set.

Problem 4-1. [30 points] **Hashing Practice**

(a) [10 points] Insert integer keys $\{16, 34, 27, 55, 93, 11, 14, 41\}$ into a hash table with eight empty slots and multiplication hash function $h(k) = (11 * k) \mod 8$, where collisions are resolved by chaining. Draw a picture of the hash table after each insertion.

(b) [10 points] Suppose you want to store n keys into a hash table of size m. Show that if the set of possible keys $[1, N]$ is sufficiently large, so that $(n-1)m < N$, then for any choice of hash function, there will always exist a set of n keys that all hash to the same location.

(c) [10 points] A *ballot* is a voter's preferred ranking of c electoral candidates, represented as c digits of a base-c number. For example, for an election between 10 candidates, the ballot 5912746083 means this particular voter thinks the eighth candidate is best and the second candidate is worst. The election board wants to construct a hash table of possible ballots to find voters that have the same preference. Show why a division hash function $h(k) = k \mod m$ is a very bad choice of hash function if m is chosen to be $c - 1$.

Problem 4-2. [40 points] **Linear time**

Show how to solve each problem in either worst case or expected $O(n)$ time.

(a) [10 points] Count the number of repeated numbers in an array of n integers. Example: $\{1, 3, 4, 8, 3, 3, 4\}$ has three repeated numbers.

(b) [10 points] Sort a set of n integers, where each integer x in the set is guaranteed to satisfy $6005n^3 < x \leq 6006n^3$.

(c) [10 points] Given a set of n integers, determine whether any pair of them sum to a given value k. Example: $\{2, 3, 4, 8, 3, 3, 4\}$ contains one pair of numbers that sum to 10, but no pair of numbers that sum to 9.

(d) [10 points] Let $\mathcal{W}$ be the set of English words. A *substitution cipher* is any bijective mapping $\mathcal{C} : \mathcal{W} \to \mathcal{W}$. To encode a message, simply replace each word w with word $\mathcal{C}(w)$. You and your friend agree on a substitution cipher to encrypt all of your correspondance. Given a long encoded message from your friend (i.e. containing $n = \Omega(|\mathcal{W}|)$ words) decode the message. Keep in mind that $|\mathcal{W}| \ll 26^s$ where s is the length of the longest English word.

Problem 4-3. [30 points] DNA Search

DNA can be represented as a sequence of four letters from the set $\{C, G, T, A\}$. For simplicity, assume the letters are encoded as integers, such that $C = 0, G = 1, T = 2$, and $A = 3$. Certain sequences of letters are known to be undesirable because they result in a genetic disease. For example, the presence of the sequence (C, A, T, T, A, G) might result in an organism that chases kittens.

Dr. Crancis Frick has a theory that it's not just a specific sequence of letters, but rather any permutation of a disease substring could also cause the disease. For example, (G, A, T, A, C, T) might also induce kitten chasing. Given a DNA sequence containing n letters and a potential disease marker containing d letters, help Dr. Frick evaluate if the sequence contains a sub-sequence that is a permutation of the queried disease marker in worst case (or expected) $(d + n)$ time.

(a) [10 points] **Frequencies:** One way to solve this problem is to keep track of how often characters in the alphabet have been seen. Show how to solve Dr. Frick's problem using a frequency table.

(b) [15 points] **Rolling Hash:** If keys are sequences in $[1, \sigma]^r$, i.e. sequences of r characters from an alphabet of size σ, a *rolling hash* function:

$$h_p(k) = \sum_{i=0}^{r-1} k[i] \cdot \sigma^i \mod p$$

satisfies that collisions occur with probability $\Pr_p\{h_p(x) = h_p(y)\} \leq 1/r$ for any $x, y \in [1, \sigma]^r$, for a suitably random choice of a prime p. This is a useful because rolling hashes can be computed from the rolling hashes of similar keys in $O(1)$ time, and two hashes can be compared in $O(1)$ time. Show how to solve Dr. Frick's problem using a rolling hash.

(c) [5 points] **Compare:** Suppose we wanted to do the same analysis on Martian DNA, containing many more types of letters. If the a letters of the Martian alphabet are encoded by the integers $[1, a]$, how would the running time of the preceding two algorithms change with respect to a?