Warm-up

Problem 1. Consider a hash table of size $N > 1$, and the hash function such that $h(k) = k \bmod 2$ for every k. We insert a dataset S of size $n < N$. After that, what is the typical running times of GET for chaining and open addressing (as a function of n)?

Problem 2. Suppose you are given a hash function h mapping 10-digit strings to integers in $\{1, 2, \ldots, 10000\}$. Show that there is some dataset S of size $1,000,000$ such that all keys of S are hashed to the *same* value.

Problem 3. Work out the details of implementing cycle detection in cuckoo hashing based on the number of iterations of the eviction sequence.

Problem 4. Work out the details of implementing cycle detection in cuckoo hashing based on keeping a flag for each entry.

Problem solving

Problem 5. Design a sorted hash table data structure that performs the usual operations of a hash table with the additional requirement that when we iterate over the items, we do so in the order in which they were inserted into the hash table. Iterating over the items should take $O(n)$ time where n is the number of items stored in the hash table. Your data structure should only add $O(1)$ time to the standard put, get, and delete operations.

Problem 6. Given an array with n integers, design an algorithm for finding a value that is most frequent in the array. Your algorithm should run in $O(n)$ expected time.

Problem 7. A multimap is a data structure that allows for multiple values to be associated with the same key. The method GET(k) should return all the values associated with key k. Describe an implementation where this method runs in $O(1 + s)$ expected time, where s is the number of values associated with k.

Problem 8. Suppose that you have a group of n people and you would like to know if there are two people that share a birthday. Design an $O(1)$ time algorithm that given the information about the n people's birthdays, finds a pair that shares a birthday, or reports that no such pair exists.

Problem 9. In computational linguistics, texts (such as a book or an article) are modelled as a sequence of words. A k-gram is a sequence of k consecutive words. A common task in language modelling requires that we compute the frequency of all k-grams that appear in the text.

Given a text with n words, design an $O(n)$ expected time algorithm that computes the frequency of all k-grams that appear at least once in the text.