Your name
CS 261, Spring 2021
Homework 3
Due Monday, April 19

1. Suppose that we wish to represent sets of n elements, all of which can be represented as k-bit binary numbers (that is, integers in the range from 0 to $2^k - 1$). We are considering the following three representations: a hash table whose cells each store a single element, using linear probing with load factor 1/2, a bitmap, or a Bloom filter with expected false positive rate ε.

(a) Write formulas for the number of bits needed to store each of these three representations. (Do not count the storage needed to represent their hash functions.)

(b) As a function of k, for which values of n does the hash table use less space than the bitmap, and for which values of n does the bitmap use less space?

(c) As a function of k, for which values of ε does the Bloom filter use less space than the hash table, and for which values of ε does the hash table use less space?

■

2. Suppose A, B, and C are sets, all represented as bitmaps. The majority set $\text{maj}(A, B, C)$ consists of all elements that belong to two or more of A, B, or C. Write an expression (like the expressions on the slide "Implementation for small universes" of the lecture notes) for $\text{maj}(A, B, C)$.

■

3. Suppose we wish to maintain a set of n items and a numbering of the items (a function mapping the items to a small range of integers), so that each two items in the set are mapped to distinct numbers, and so that we can look up the number for each item in the set in constant time. If the set changes, the numbering can also change. Describe how to do this using a cuckoo-hash based set representation, without using any more storage space than the set itself would already use.

■

4. The lecture notes included a lower bound showing that (under standard assumptions) it is not possible to handle disjoint-set queries in substantially sublinear time per query, using a data structure whose construction takes polynomial time and space. However, this does not rule out fast queries using bigger and slower-to-construct data structures.

Suppose that

- We are given as input a family of n sets S_i, each of size k.
- Each set is a subset of the numbers from 0 to $N - 1$ for some N (either because they were given that way or because we used the answer to question 3 to number the set elements)
- We wish to handle disjoint-set queries of the following form: given a query set Q of size k, as a list of its elements, answer "yes" if the input sets include a set S_i that is disjoint from Q, or "no" if no such set exists. When the answer is yes, it is not necessary to find which set S_i is disjoint.

Describe how to use bitmaps to represent a set-of-sets data structure that can answer disjoint-set queries in time $O(k)$, using a data structure whose space and construction time is exponential in N. You may assume that operations on N-bit binary numbers take constant time per operation.

■