Fundamental Algorithms, Section 003
Homework 7, Additional Problems, Fall 22.

1. This problem concerns a variant of Bloom filters where there are two disjoint sets, a threat set T and a safe set S; these two sets are subsets of a universe U. Let $s = |S|$ and $t = |T|$. We want to ensure that the correct membership answer is given for both sets in the following situation: $s^2 \le t$.

We will view all of $U \setminus T$ as being safe, but it is only on S that we want to be certain of a correct answer. Still, we will want the probability of an incorrect answer on $U \setminus (S \cup T)$ to be at most $1/2^{m_t - 1}$, where m_t is the number of hashes the Bloom filter on T uses.

We will construct a Bloom filter for both sets, however the Bloom filter for set S will use space $4m_s \cdot t$, where m_s is the number of hashes being performed.

You may assume the probability of a wrong answer on the Bloom filter for T is at most $1/2^{m_t}$ (in the case that the item is not in T and m_t is the number of hashes in this filter), and the probability of a wrong answer on the Bloom filter for S is at most $(s/2t)^{m_s}$.

Suppose you have a family of hash functions for both sets from which you can make uniform random draws. Show how to construct the two Bloom filters so you can always obtain the correct answer on both sets. Your construction should take expected time $O(s+t)$.

What is the probability that the Bloom filter on S never reports an item $x \in T$ being in S? How can you ensure the number of such wrong reports is 0? Your algorithm may use expected $O(t)$ time.

Comment. The space bound for the Bloom filter for S holds when using the universal hash functions analyzed in problem 1 on homework 7.

2. This problem investigates the cost of rebuilding hash tables when their size changes substantially for hashing with chaining.

Suppose we are implementing a dictionary supporting search, insertion and deletion queries. Suppose we have families $\mathcal{H}_i = \{h_{i1}, h_{i2}, \ldots\}$ of hash functions for hash tables of sizes 2^i for $i =, 1, 2, 3, \ldots$.

Suppose that when a hash table of size 2^i is full, i.e. it holds 2^i items, we put the items in a hash table of size 2^{i+1} by drawing a hash function uniformly at random for the new table, and rehashing all 2^i items at hand. Likewise, whenever a table of size 2^i is 1/4 full, i.e. it holds $2^i/4$ items, we move all the items, 2^{i-2} of them, into a table of size 2^{i-1}. At all times, there will be one table in use.

Suppose that whenever a new table of a given size is requested that it is provided uninitialized in $O(1)$ time.

Show that if we start with an empty table of size 2 and perform a sequence of m insertions and deletions, possibly intermixed with searches, then the cost of the m insertions and deletions is expected $O(m)$ time.

Hint. What is the minimum number of insertions or deletions from the time a table of size 2^i is creates to the time we create a different table (either of size 2^{i+1} or of size 2^{i-1})? Also, what is the cost for created a new table from the table of size 2^i?