6.897 Advanced Data Structures (Spring'05)

Prof. Erik Demaine TA: Mihai Pătraşcu

$\boxed{\text{Problem 1}}$ *Due: Monday, Feb. 7*

Be sure to read the instructions on the assignments section of the class web page.

Preliminaries. Assume we have two uniformly random hash functions $h_1, h_2 : U \to \{1, 2, \ldots, cn\}$. (Alternatively, assume h_1, h_2 are n-wise independent.) You can choose c to be a sufficiently large constant. Ignore the space and time needed to choose random h_1 and h_2, and assume that they can be evaluated in constant time.

Remember that cuckoo hashing simply holds an array $T[1\,..\,cn]$ of keys, and maintains the property that any $x \in S$ is either in $T[h_1(x)]$ or $T[h_2(x)]$. As we did for the analysis of cuckoo hashing, consider the graph G with vertex set $\{1, 2, \ldots, cn\}$ and edge set $\{(h_1(x), h_2(x)) \mid x \in S\}$.

Prove the following lemma: If h_1, h_2 are chosen to be uniformly random hash functions, then with probability at least $\frac{1}{2}$, the graph G contains no cycles.

Hint: look at the analysis from Lecture 1 for cuckoo insertions, in the "two cycles" case.

Bloomier filters. Now consider the static Bloomier filter problem, defined as follows. We are given a static set S, $|S| = n$, and we associate with every value in S an r-bit quantity. A query must return the data associated with a given $x \in S$. It is guaranteed that a query is given a value in S (otherwise, the behavior of a query can be arbitrary).

Prove the following: Using the lemma from above, construct a static Bloomier filter using $O(nr)$ bits of space, which answers queries in $O(1)$ worst-case time. The construction time should be polynomial in n, in expectation.

Observe that the space can be less than n cells, so the data structure *cannot store S*!