Homework 4 - CSCI 181 - S20

1. (10 points) Suppose we have a hash function h that takes inputs of 1088 bit strings and outputs hash strings of 256 bits.

   (a) As discussed this function will have collisions, and on average, h is an n-to-1 map. Find n. (To make it easier to solve this problem, think about a similar, simpler problem and solve it first. For example suppose you have a function $g : S \to T$ where S has 6 elements and T has 2 elements.)

   (b) For an output y, we expect to have n input strings that map to it. So there will be n different 1088 bit strings x such that $h(x) = y$. If we want to solve the one-way problem for an output string y, we need to find one of the n x's among all 1088 bit strings. This seems easy to do as we only need to find ANY 1088 bit string x such that $h(x) = y$ and we have n of such x's. However, the probability that we will solve the one-way problem by applying h to random 1088 bit strings is n/(number of 1088 bit strings). Find this probability.

2. (10 points) Let $f : X \to Y$ be a hash function and assume $|X|/|Y|$ is very large (note $|X|$ and $|Y|$ are the sizes of the sets X and Y, respectively). Write an informal proof that if f has the weakly collision resistant property then f has the one-way property. You can do this by writing a contrapositive proof. So assume that f does NOT have the one-way property, and then give an informal proof that f will NOT have the the weakly collision resistant property.

   Recall: One-way property: Given $y \in Y$ it is infeasible to find $x \in X$ such that $f(x) = y$.

   Weakly collision resistant property: Given $x \in X$ it is infeasible to find $x' \in X$ with $x' \neq x$ such that $f(x') = f(x)$.

Homework 4 ends here. If you want to get a head start on the next week. Implement these two functions by writing a program. You do no need to submit these at this time, but you will need to submit them next week. You will understand the purpose of these functions later.

1. Implement a function that turns a 1-dimensional array of length 1600, $v[0 \ldots 1599]$, to a 3-dimensional array $a[0 \ldots 4][0 \ldots 4][0 \ldots 63]$ such that $a[i][j][k] = v[64(5j + i) + k]$.

2. Implement a function that turns a 3-dimensional array $a[0 \ldots 4][0 \ldots 4][0 \ldots 63]$ into a 1-dimensional array of length 1600, $v[0 \ldots 1599]$, such that $v[64(5j+i)+k] = a[i][j][k]$.