INTRO TO APPROXIMATION, CLASS 6

hashing

D: A set of random variables $X_1, \dots X_n$ is *k-wise independent* if for any subset $I \subseteq \{1, \dots, n\}$ with $|I| \leq k$ and any possible outcome values c_i, the multiplication property for independence holds:

$$Pr[\bigwedge_{i \in I}(X_i = c_i)] = \prod_{i \in I} Pr[X_i = c_i].$$

In the following we consider functions $h\colon U \to HT$ with $|U| = m$ and $|HT| = n$ (HT is a hash table).

D: A family of functions $\mathcal{H}$ is *k-universal* (we could also say *weakly k-universal*) if for any distinct elements $x_1, x_2, \dots x_k \in U$ and a hash function h chosen uniformly at random from $\mathcal{H}$, we have

$$Pr_h[h(x_1) = h(x_2) = \dots = h(x_k)] \leq \frac{1}{n^{k-1}}.$$

This means that a uniformly random h from a weakly k-universal family just needs to avoid too many hash table conflicts.

D:A family of hash functions $\mathcal{H}$ is *strongly k-universal* if for any distinct elements $x_1, x_2, \dots x_k \in U$, any (even non-distinct) values $y_1, y_2, \dots, y_k \in HT$ and a hash function h chosen uniformly at random from $\mathcal{H}$, we have

$$Pr_h[h(x_1) = y_1 \wedge h(x_2) = y_2 \wedge \dots \wedge h(x_k) = y_k] = \frac{1}{n^k}.$$

We can rephrase this as follows: A family of hash functions $\mathcal{H}$ is *strongly k-universal* if we can hash k elements with a uniformly random hash function and their hashed positions behave like we chose *the hash positions themselves* uniformly at random.

EXERCISE ONE One of the most interesting applications of universal hashing theory is derandomizing probabilistic algorithms using pairwise (or k-wise) independent random variables (bits) instead of fully independent random variables. This works because we are able to generate many pairwise independent random bits from just a few fully random bits.

Your task is to show how we can do it. Given k fully independent random bits, $x_1, x_2, x_3, \dots, x_k$, show how we can create $2^k - 1$ pairwise independent random binary variables $y_1, y_2, \dots, y_{2^k-1}$.

EXERCISE TWO Let us have k uniformly random independent bits. We define $X_{i,j}$ for $1 \leq i < j \leq k$ as the indicator whether the i-th and j-th bit are equal. Show that $X_{i,j}$ are 2-wise independent, but not 3-wise independent.

EXERCISE THREE You have seen at the lecture that the family of functions $h_{a,b}(x) = ax + b \mod p$ is a strongly 2-universal family when both the universe U and the hash table HT are of the same size.

This is quite impractical, as usually one hashes a large universe into a reasonably compact hash table. Therefore, suppose that we have $|U| = m$, $|HT| = n$ and $p \geq m$. Prove that almost the same family of functions, namely

$$h_{a,b}(x) = (ax + b \mod p) \mod n$$

is weakly 2-universal.

The proof will go as follows: for a given $x_1 \neq x_2$, we want to count the number of pairs (a, b) (or the number of hash functions) which will cause x_1 and x_2 to collide.

1. Use the strong 2-universality or a direct argument to show that for a given quadruple x_1, x_2 and $c, d \in \{0, \ldots p-1\}$, there is exactly one pair (a, b) such that

$$ax_1 + b = c \quad \wedge \quad ax_2 + b = d.$$

2. Show that instead of counting pairs (a, b) which cause collisions, we can count pairs (c, d) where for each c, d it holds that $c \neq d$ and $c = d \mod n$.
3. Finally, do the counting.

EXERCISE FOUR We know that k bits are sufficient for generating $O(2^k)$ many random pairwise-independent variables. The question now is: how many do we need for 3-wise independent variables?

Surprisingly, you can generate 2^{k-1} of them using again just k bits. Suggest a generator and prove that the result are 3-wise independent random bits.