Cpt S 515 Homework #5

No late homework!

1. Consider a family $\mathcal{H}$ of hash functions:

$$\mathcal{H} = \{h_i : 1 \leq i \leq 8\}.$$

Each h_i is to map an array of eight bits into its i-th component: $h_i(a_1 \cdots a_8) = a_i$. Is $\mathcal{H}$ universal? why or why not?

2. Here is a classic example of universal family of hash functions. Let M be a prime number and, as usual, $[M] = \{0, 1, \cdots, M-1\}$. Consider the following family of hash functions:

$$h_{\mathbf{r}}(\mathbf{x}) = \quad (\mathbf{r} \cdot \mathbf{x} \bmod M),$$

where $\mathbf{r}, \mathbf{x} \in [M]^k$ (where k is a given constant like 10), and $\mathbf{r} \cdot \mathbf{x} = \sum_i r_i x_i$. Show that the family of hash functions (for the given k) is universal.

3. So far, what we have learned about hasing is to hash an array of numbers into one number (e.g., locality sensitive hashing). Can you suggest a way to hash a graph into a number (which could be a real number)?

4. Randomized quicksort is a Las Vegas algorithm where the first step is to create a random permutation of the input arrayof numbers before the second step of running quicksort. Now, we assume that we have a high quality psuedo random generator $r(n)$ that will generate a random number in $1..n$. Please show how to generate a "random" graph with 5 nodes.

5. Mr. X drives on I-90 all the way from Pullman to New York (Let's assume that Pullman is Spoakne). On his car, there is a device that can suggest all the interesting places nearby that Mr. X might visit (and spend some money at these places of course). These places are stored in a set S and will be updated automatically while Mr. X is driving. Please suggest a way to implement the S so that Mr. X can query (e.g., "Is there a restrant nearby?", etc.). You shall use Bloom filter to store S. Feel free to look up papers on the Internet.

6. We know many ways to hash an array of integrs into a number. However, hash itself is loosy — that is, the function many not be one-to-one. Can you suggest a way to hash an array of 10 bits into a number such that a. the hash is one-to-one, and, b. the hash is locality sensitive (i.e., when the Hamming distance between two such arrays of 10 bits is small, then so is the distance between their hash values). (I have a terrific way to do this — but I wont tell you. You shall figure out your own ways to do this. This problem concerns a lot of fundamental applicational problems in computer science.)