# COS 341: Discrete Mathematics

Precept #7 Fall 2006
For the week of: November 27

---

**1.** A *tournament* is a directed graph in which, for every pair of distinct vertices $i$ and $j$, there is either an edge directed from $i$ to $j$, or an edge directed from $j$ to $i$, but not both. A *Hamiltonian cycle* is a directed path through a graph that starts and ends at the same vertex, and visits every vertex in the graph exactly once.

a. Find the expected number of Hamiltonian cycles in a random $n$-vertex tournament. (Here, "random" means that, for each pair of distinct vertices $i$ and $j$, each of the two possible edges $i \to j$ and $j \to i$ is included with equal probability, independent of all other pairs of vertices.)

b. Show that there exists a 10-vertex tournament with over 300 Hamiltonian cycles.

**2.** A new research project at Princeton is attempting to build an index of documents with the capability to do fast similarity searches. The raw representation of each document in this database is a 10,000 bit string. The similarity of two documents is the fraction of bits their representations agree in.

An important task for this system is to compare pairs of documents and decide if they are more than 95% similar or less than 90% similar. However comparing the raw representations takes a very long time. The cool new idea is to produce a short 100 bit hash value for each document such that the similarity of two documents can be approximated from examining their signatures. In this exercise, we will evaluate two schemes for producing such hash values.

Let $x_i$ denote the bit in position $i$ of $x$.

**Scheme 1:** Pick 100 random positions $i_1, ..., i_{100}$ (with repetitions allowed). The hash value for raw representation $x$ is $h_1(x) = x_{i_1} x_{i_2} \dots x_{i_{100}}$. (Note that the set of 100 random positions is picked once and all hash values are computed according to it.)

**Scheme 2:** Pick $100k$ random positions

$$\begin{matrix} i_1^1 & i_1^2 & \dots & i_1^k \\ i_2^1 & i_2^2 & \dots & i_2^k \\ \vdots & \vdots & \ddots & \vdots \\ i_{100}^1 & i_{100}^2 & \dots & i_{100}^k \end{matrix}$$

Now let

$$r_s(x) = \bigoplus_{t=1}^{k} x_{i_s^t} = x_{i_s^1} \oplus x_{i_s^2} \oplus \dots \oplus x_{i_s^k}.$$

In other words, let $r_s(x)$ be equal to 0, if there is an even number of ones among $x_{i_s^1}$, $x_{i_s^2}$, ..., $x_{i_s^k}$; and let $r_s(x)$ be equal to 1 otherwise. The hash value for raw representation $x$ is $h_2(x) = r_1(x) r_2(x) \dots r_{100}(x)$.

a. Assume that the similarity of $x$ and $y$ is $p$ (where $0 \le p \le 1$). Compute the probability that the first bit of the hash value of $x$ is different from the first bit of the hash value of $y$. Compute this probability for the first and second schemes.

b. Again, assume that the similarity of $x$ and $y$ is $p$ (where $0 \leq p \leq 1$). Compute the expected number of bits the hash values of $x$ and $y$ differ in. Compute this number for the first and second schemes.

   Denote this number for the second scheme by $d(p)$.

c. For what value of $k$ is the difference between $d(0.95)$ and $d(0.9)$ maximal?