MATH 642:582–Fall, 2003

Assignment 2–Due October 3 (Version: September 18)

1. Fix a positive integer p. Find an exact expression for the number of permutations of $[n]$ that have no cycle of length exactly p. (Your expression will be a sum, but it should be as simple as possible). Determine the asymptotics of your expression as n tends to ∞.

2. Prove that the degree sequence of a graph on more than four edges is edge-reconstructible. More precisely, for any graph G with $m \geq 4$ edges, if you are given a multiset consisting of the m unlabeled graphs on $m-1$ edges obtained by deleting each edge of G in turn, then the degree sequence of G can be completely determined. (This problem does not use inclusion-exclusion, just thinking.)

3. Jukna, Problem 3.3

4. Jukna, Problem 3.7

5. (a) Consider a finite probability space Ω, and subsets (events) $A_1, A_2, \ldots, A_n$. As usual, for $I \subseteq [n]$, define $A_I = \cap_{i \in I} A_i$ and let $p_I = prob(A_I)$. Let $B_1, B_2, \ldots, B_n$ be another set of events and define $q_I = prob(B_I)$. Suppose that $p_I = q_I$ for all $I \neq [n]$ but $p_{[n]} \neq q_{[n]}$. Prove that Ω must have at least 2^{n-1} elements.

   (b) Give an example to show that the bound in the first part is best possible.

6. A *Hamiltonian path* in a graph $G = (V, E)$ is a permutation of the vertices $v_1, \ldots, v_n$ such that each pair v_i, v_{i+1} is adjacent in G for $1 \leq i < n$. A naive algorithm for testing whether a graph has a Hamiltonian path requires checking all possible permutations of the vertex set, and thus requires $n!poly(n)$ number of steps, where $poly(n)$ denotes an unspecified polynomial function of n. The purpose of this problem is to derive an algorithm that runs in $2^n poly(n)$ steps (and also requires memory size only $poly(n)$).

   (a) A walk in a graph is a sequence $v_1, \ldots, v_k$ of vertices, possibly with repitition such that each pair v_i, v_{i+1} is adjacent in the graph. For vertices x, y and integer k let $W_G(x, y, k)$ denote the number of k step walks from x to y. Give a method for computing $W_G(x, y, k)$ that uses at most $poly(n, k)$ steps.

   (b) Let $H_G(x, y)$ denote the number of Hamiltonian paths in G. Use the answer to the previous problem and inclusion-exclusion to derive a formula for computing $H_G(x, y)$ that uses at most $2^n poly(n)$ computation steps. (Without being too precise here about what a "computation step" is, we assume that adding, subtracting or multiplying two k bit numbers can be done in $poly(k)$ computation steps.