PROBLEM SET 5

1. This problem asks you to work through the steps of constructing a binary de Bruijn string 16.

Let D_3 be the digraph whose vertices are the length 3 binary strings, and where there is an edge from *abc* to *bcd*.

1.a Draw D_3 .

1.b Draw a spanning arboresence of D_3 , rooted at 000.

1.c Follow the BEST algorithm to find a Eulerian tour in D_3 .

1.d What is the resulting de Bruijn cycle?

2. The graph $K_{p,q}$ has p + q vertices and pq edges. Specifically, the vertices are called $a_1, a_2, \ldots, a_p, b_1, b_2, \ldots, b_q$ and there is an edge between a_i and b_j .

2.a Write down the adjacency matrix and the Laplacian of $K_{2,3}$.

2.b Let $M_{p,q}(x, y)$ be the matrix obtained by taking the adjacency matrix of $K_{p,q}$, changing the first p diagonal entries to x's and q diagonal entries to y's. For example, here is $M_{3,2}(x, y)$:

$$\begin{pmatrix} x & 0 & 0 & 1 & 1 \\ 0 & x & 0 & 1 & 1 \\ 0 & 0 & x & 1 & 1 \\ 1 & 1 & 1 & y & 0 \\ 1 & 1 & 1 & 0 & y \end{pmatrix}$$

Show that det $M_{p,q}(x,y) = x^p y^q - pq x^{p-1} y^{q-1}$. (Hint: write $M_{p,q}(x,y)$ as a sum of simpler matrices.)

3. Let G be the directed graph which has n vertices arranged around a circle and has 4n edges, such that, for every adjacent pair of vertices (u, v), there are two edges $u \to v$ and two edges $v \to u$.

3.a Count the number of spanning arboresences of G using the matrix tree theorem.

3.b Describe all of these arboresences explicitly.

3.c Determine the number of Eulerian tours in G.

4. Let G be a connected graph where every vertex has even degree. We define an *Eulerian* orientation of G to be a way of making G into a directed graph so that every edge has the same indegree and outdegree. Show that G has an Eulerian orientation.

5. This problem continues the terminology from problem 4. We will also need to introduce another piece of terminology: Let H be a *bipartite graph*. This means that the vertices of H can be divided into two sets X and Y, so that all edges of H have one endpoint in X and one in Y. Suppose furthermore that |X| = |Y|. A *perfect matching* of H is a collection M of edges of H so that every vertex of H is in precisely one edge of M.

The aim of this problem is to show that, if we could count Eulerian orientations, we could count perfect matchings. A classic paper of computer science¹ shows that counting perfect matchings is as hard as any counting problem can be. (Technical term: It is #P complete.) Thus, this is evidence that there is no good method to count Eulerian orientations.

5.a Let H be a bipartite graph, with X and Y the two classes of vertices of H. For simplicity, assume that every vertex of H has degree ≥ 2 and |X| = |Y|. Let G_0 be the graph defined as follows: Take H and add two more vertices a and b. For $x \in X$, add $d_x - 2$ edges from x to a. For $y \in Y$, add $d_y - 2$ edges from y to b.

Show that $\deg a = \deg b$. We will call this number D.

5.b Consider any orientation of G_0 where every vertex in X or Y has equal indegree and outdegree. Show that the number of edges directed into a is the same as the number of edges directed out of b.

5.c Let E_{ℓ} be the number of orientations of G_0 where every vertex in X or Y has equal indegree and outdegree, and where there are ℓ edges into a and ℓ edges out of b.

Show that E_0 is the number of perfect matchings of H.

5.d Let G_k be the graph G_0 with k additional edges added between a and b. Suppose that k = D + 2m, for some integer m.

Express the number of Eulerian orientations of G_k as a linear combination of E_0, E_1, \ldots, E_D .

If we had a fast way of counting Eulerian orientations, we could solve the linear equations in part 5.d and compute the E_{ℓ} 's. In particular, we could compute E_0 , which is as hard as any counting problem.

A paper of Brightwell and Winkler² uses a similar, but more complex, argument to show that, if we could count Eulerian tours efficiently, we could count Eulerian orientations efficiently. Thus, there is strong reason to believe there is no way to count Eulerian tours.

¹Leslie G. Valiant, "The Complexity of Computing the Permanent". *Theoretical Computer Science* 8: 189–201 (1979)

²Brightwell and Winkler, "Note on Counting Eulerian Circuits", CDAM Research Report LSE-CDAM-2004-12, (2004)