This problem set has a total of 12 problems, and was last updated on Feb 15, 2017.
You can do as many as you like, and either discuss them with me or the TA during office hours, or submit a written solution (which is much harder to grade than some interactive discussion).
Throughout this problem set, you may assume that matrix products, inverses and ranks (over reals, or finite fields) can be computed in $O(n^\omega)$ time, and all arithmetic operations involving real numbers take $O(1)$ time.

1. Recall the Edmond’s matrix of a bipartite graph as defined in class for bipartite graphs $G = (V, E)$ where $V$ has bipartition $V = A, B$:

   $M_{ab} = \begin{cases} x_{ab} & \text{if } ab \in E \\ 0 & \text{otherwise} \end{cases}$.

   Show that for any bipartite graph (where $A$ and $B$ can be of different sizes), the rank of this matrix equals the size of the maximum matching.

2. Suppose we can multiply $n \times n$ matrices in $f(n)$ time such that $f(n) > n^2$.

   Show that given an $n \times n$ matrix $A$, we can compute its inverse, $A^{-1}$ in $O(f(n) \log n)$ time. **HINT:** use the more general definition of Schur complements that remove many vertices.

3. Give an $O(n^\omega)$ time algorithm for extracting a perfect matching from a bipartite graph. (hint: use Schur complements).

4. In class we talked about the number of spanning trees of a ‘weighted complete graph’.

   Formally such a graph is given by a set of weights $w_1 \ldots w_n$, one per vertex, and the weight of the edge between vertices $i$ and $j$ is $w_i \times w_j$.

   Show, using Prufer code, or any other method of your choice that the total weight of spanning trees of this graph is

   $\left( \sum_{i=1}^{n} w_i \right)^{n-2} \cdot \prod_{i=1}^{n} w_i$. 

Note that the weight of a tree, $T$, is defined to be the product of the edge weights in it:

$$w(T) \overset{\text{def}}{=} \prod_{e \in T} w_e,$$

and total weight of trees is $\sum_T w(T)$.

5. (due together with Problem Set 2 due to misleading hint) The effective resistance of an edge is defined as

$$r_e = \chi_e L^\dagger \chi_e$$

where:

(a) $L$ is the graph Laplacian matrix,
(b) $\chi_e$ is the indicator vector with 1 on one endpoint of $e$, and $-1$ on the other.
(c) $\dagger$ denotes the pseudoinverse: $L$ may not be full rank, but one can show that $\chi_e$ is in its image space, so the linear system $Lx = \chi_e$ still has a solution (which is unique if we restrict to solutions orthogonal to the null space).

Show that for an unweighted graph (this is not crucial, it just makes some of the normalization easier) that the fraction of spanning trees that use $e$ is exactly $r_e$.

**HINT:** use Cramer's rule (the adjugate matrix [https://en.wikipedia.org/wiki/Adjugate_matrix]) of $L_{1:(n-1),1:(n-1)}$, aka $L$ with the last row/column removed.

6. Show the following more general version of the matrix-tree theorem: let $L$ be the Laplacian of some graph $G$, and let $M$ be the matrix obtained by removing any row, and any column from $L$ (note that the row id may be different than the column id). Then

$$|\det(M)| = T(G).$$

You may assume the matrix tree theorem in your proof.

7. Show that a connected $d$-regular undirected, unweighted graph is bipartite if and only if its adjacency matrix has $-d$ as an eigenvalue.

**HINT:** show that a $d$-regular graph cannot have an eigenvalue less than $-d$; then pull some structure out of the equality conditions.

8. Call a graph $G$ a **banana plantain**\(^1\) if its Laplacian, $L_G$ has:

(a) All $n$ eigenvalues distinct, and

---

\(^1\)This is not an actual term, Richard made it up so he doesn’t have to write this twice for $G$ and $H$, and as it turned out, there are actually banana graphs: [https://arxiv.org/abs/0807.1690](https://arxiv.org/abs/0807.1690).
(b) Aside from the all-1s eigenvector, all \( n \times (n - 1) \) entries in the unit eigenvectors \( \mathbf{u}_2 \ldots \mathbf{u}_n \) have distinct absolute values.

Given two plantain graphs \( G \) and \( H \), determine whether they are isomorphic in \( O(n^{10}) \) time. **HINT:** apply the spectral theorem and work on the resulting diagonalization.

9. The purpose of this exercise is to demonstrate that escape probabilities are not ‘numerically robust’. Exhibit an undirected graph \( G \) and a source/sink pair \( s \) and \( t \), as well as a start vertex \( u \), along with a perturbation of edge weights in \( G \) by factors between \([1, 2]\) that changes the escape probability from \( u \) by a factor of more than 10.

10. Prove or disprove: in a weighted, undirected graph, perturbing only the weight of a single edge by a factor of 2 can only change the escape probability from \( u \) (arriving at \( t \) before \( s \)) by a factor of up to 10.

11. In a connected graph \( G \), let \( \mathbf{x} \) be the voltage needed to send an electrical flow meeting demands \( \mathbf{b} \) that’s not zero everywhere. Show that the minimum and maximum values of \( \mathbf{x} \) are attained at vertices \( u \) where \( \mathbf{b}_u \neq 0 \).

12. Consider the \( n \times n \) 2-D square grid with undirected, unit weighted edges. Show that the effective resistance between the bottom left corner and the top right corner is bounded by \( \Theta(\log n) \). (you need to provide both lower and upper bounds)