CS591: Graph Analytics **Release: Feb. 17th, 2020**

Homework 1 Due by March 16th, 2020

Prof. Charalampos E. Tsourakakis *Deadline: March 16th, 2020*

[0] Some graph-related preliminaries

(A) We consider the following game on an undirected, simple graph G. There are two players, a red color player R and a blue color B. Initially, all edges are uncolored. Player R starts playing first, and the two players alternately color an uncolored edge of G with their color. The goal of B is that in the end of the game the blue colored edges form a connected spanning subgraph of G. The goal of R is to prevent this.

- Show that B can always win if G contains two edge-disjoint spanning trees.

(B) Let $k \geq 2$. Show that in a k-connected graph any k vertices lie on a common cycle. [Hint: One idea is to proceed by induction on k.]

(C) Given positive integers $d_1, \ldots, d_n$ such that $\sum_{i=1}^{n} d_i = 2n-2$, how many labeled trees with vertex set $\{1, \ldots, n\}$ are there such that vertex i has degree d_i for each i? [Hint: Review Prüfer's code.]

(D) Let T be a tree on $k \geq 2$ vertices. Let G be a graph whose minimum degree $\delta(G)$ satisfies $\delta(G) \geq k-1$.

- Prove or disprove: Does G always contain a copy of T?

(E) Prove that the Ford-Fulkerson algorithm terminates for rational capacities.

(F) Deduce Hall's marriage theorem from the max-flow/min-cut theorem.

(G) An $n \times n$ matrix with entries from $\{1, \ldots, n\}$ is called a *Latin* square, if every element of $\{1, \ldots, n\}$ appears exactly once in each column, and exactly once in each row. Recast the problem of constructing Latin squares as coloring problem.

(H) Let $G(V, E), |V| \geq 2$ be a connected weighted graph with distinct positive integer weights on the edges.

- Prove or disprove: For every vertex v, the edge e of minimum weight that is incident to v is necessarily in any minimum spanning tree (MST) of G.
- Prove or disprove: G has a unique MST.

[1] Probability

(A) Cauchy-Schwartz inequality Prove the Cauchy-Schwartz inequality for random variables X, Y

$$\boxed{|\mathbb{E}[XY]| \leq \sqrt{\mathbb{E}[X^2]}\sqrt{\mathbb{E}[Y^2]}.}$$

(B) Bonferonni Inequalities Let $E_1, E_2, \ldots, E_n$ be events in a sample space. We have been using the union bound a lot in our class:

$$\mathbf{Pr}[E_1 \cup \ldots \cup E_n] \leq \sum_{i=1}^{n} \mathbf{Pr}[E_i].$$

In this exercise you will prove a more general result. Define

$$S_1 = \sum_{i=1}^{n} \mathbf{Pr}[E_i]$$
$$S_2 = \sum_{i<j} \mathbf{Pr}[E_i \cap E_j]$$

and for $2 < k \leq n$,

$$S_k = \sum_{(i_1,\ldots,i_k)} \mathbf{Pr}[E_{i_1} \cap \ldots \cap E_{i_k}],$$

where the summation is taken over all ordered k-tuples of distinct integers.

<u>Prove</u> for *odd* k, $1 \leq k \leq n$

$$\boxed{\mathbf{Pr}[E_1 \cup \ldots \cup E_n] \leq \sum_{j=1}^{k} (-1)^{j+1} S_j.}$$

and for *even* k, $2 \leq k \leq n$

$$\boxed{\mathbf{Pr}[E_1 \cup \ldots \cup E_n] \geq \sum_{j=1}^{k} (-1)^{j+1} S_j.}$$

(C) Chernoff bound Let's assume that we have a biased coin such that $\mathbf{Pr}[Heads] = 0.6$. Assuming all coin tosses are independent, how many tosses n do I need to be sure with 99% probability that no less than $0.55n$ tosses are heads?

(D) Poisson Limit Theorem Recall that a random variable Z has a Poisson distribution with parameter λ, denoted

$$Z \sim Po(\lambda),$$

if it takes values in $\{0, 1, \ldots\}$ with probabilities

$$\mathbf{Pr}\left[Z = k\right] = e^{-\lambda}\frac{\lambda^k}{k!},\ k = 0, 1, 2, \ldots$$

Let $X_{N,i}$ $1 \leq i \leq N$ be independent random variables $X_{N,i} \sim Ber(p_{N,i})$, and let $S_N = \sum_{i=1}^{N}$. Assume that as $N \to +\infty$, $\max_{i \leq N} p_{N,i} \to 0$ and $\mathbb{E}\left[S_n\right] \to \lambda < +\infty$. Then, as $N \to +\infty$,

$$S_N \to Po(\lambda) \text{ in distribution.}$$

[3] Erdös-Rényi graphs

(A) Practicing the first moment method Let $G \sim G(n, p)$ where $p = o(n^{-3/2})$. Prove that G consists of isolated vertices and independent edges.

(B) Cycles in $G(n, p)$ Prove that the threshold for the emergence of cycles in $G(n, p)$ is $p^* = \frac{1}{n}$.

(C) Perfect matchings in random bipartite graphs $B(n, n, p)$ Let $p = \frac{\log n + c}{n}$ where c is a constant. Let G be a random subgraph of the complete bipartite graph $K_{n,n}$. given by taking each edge with probability p, where choices are made independently. Show that

$$\mathbf{Pr}\left[G \text{ has a perfect matching}\right] \to e^{-2e^{-c}}$$

as $n \to +\infty$.

[Hints: (a) Use the Bonferroni inequalities to “sandwich” the probability of the event “no vertex is isolated”. (b) Then, prove that the main reason why there can be no perfect matching in G are isolated vertices. In other words, show that the probability that Hall’s theorem is violated for any other reason is $o(1)$.]

[4] Not a Small world

Consider a grid of n^2 points in 2 dimensions with each node connected to its four nearest neighbors (i.e., 2d grid). In addition, each node i chooses another node j uniformly at random and established an undirected connection to it. Our goal is to route a message from a source node s to a target node t with $r(s, t) \geq \frac{n}{2}$. Here $r(s, t)$ is the distance between s, t on the grid (i.e., without the random edges).

- Prove that the expected number of steps required by *any* decentralized algorithm is at least $\sqrt{n}$.

[5] Empirical Properties of Networks

In this problem you will study empirically various properties of networks[1]. First, download the following graphs[2]

1. Amazon product co-purchasing network from March 2 2003 from
`http://www.cise.ufl.edu/research/sparse/matrices/SNAP/amazon0302.html`
2. Arxiv High Energy Physics paper citation network from
`http://www.cise.ufl.edu/research/sparse/matrices/SNAP/cit-HepPh.html`
3. Road network of Pennsylvania from
`http://www.cise.ufl.edu/research/sparse/matrices/SNAP/roadNet-PA.html`
4. Web graph of Notre Dame from
`http://www.cise.ufl.edu/research/sparse/matrices/SNAP/web-NotreDame.html`
5. Gnutella peer to peer network from August 9 2002 from
`http://www.cise.ufl.edu/research/sparse/matrices/SNAP/p2p-Gnutella09.html`.

You may use your favorite programming language to code up the following tasks. You may re-use existing software (actually, you should). Check the Web page under the Resources tab to find links to useful packages.

(A) For each graph: if it is directed, make it undirected, by ignoring the direction of each edge. Remove multiple edges and self-loops.

(B) For each graph:

- Report the number of vertices and edges. Compute the average degree and the variance of the degree distribution.
- Generate the following frequenty plot: the x-axis will correspond to degrees and the y-axis to frequencies. The function you will plot is $f(x) = \#$vertices with degree x. Re-plot the same function in log-log scale.
- Use the code available at `http://tuvalu.santafe.edu/~aaronc/powerlaws/` to fit a power-law distribution to the degree sequence of the graph. Report the output of the *plfit* function.

(C) Plot a histogram of the sizes of the connected components of each graph.

(D) For each graph, pick any vertex v in the connected component of the largest order. Report the id of the vertex you chose and compute for each $k = 1, 2, ..,\quad f(k) = \#$ vertices at distance k from v. Plot $f(k)$ versus k.

(E) For each graph compute the diameter of the largest connected component.

[1] *Send me your code by e-mail.*
[2] The files are .mat. If you are not using MATLAB you can download the same graphs in different format from `http://snap.stanford.edu/data/`.

(F) For each graph:

1. Compute for each vertex v in how many K_3s it participates in.
2. Compute the local clustering coefficients and plot their distribution.
3. Let k=degree, $f(k)$ =average number of triangles over all vertices of degree k. Plot $f(k)$ versus k in log-log scale, including error bars for the variance. Fit a least squares line and report the slope.
4. How can you use the previous answer to find outliers in a network?

(G) For each graph report the top-20 eigenvalues of the adjacency matrix.

(H) For each of the five (5) graphs, generate a random binomial graph on the same number of vertices n_i, where n_i is the number of vertices in G_i, $i = 1, \ldots, 5$ with $p = \frac{2 \log n_i}{n_i}$. Answer questions (A) through (G) for these graphs.

(I) Make a high-level evaluation of your findings. For instance, how different is the road network from the Web graph? Also, compare your findings between real-world networks and random binomial graphs.