CME 305: Discrete Mathematics and Algorithms

Instructor: Reza Zadeh (rezab@stanford.edu)

HW#1 – Due at the beginning of class Thursday 01/26/17

1. Prove that at least one of G and $\overline{G}$ is connected. Here, $\overline{G}$ is a graph on the vertices of G such that two vertices are adjacent in $\overline{G}$ if and only if they are not adjacent in G.

   Solution: Let G be a disconnected graph in which case we can decompose it into k connected components $C_1, C_2, \ldots, C_k$. We want to show that $\overline{G}$ is connected i.e. there is a path between any u and v in $\overline{G}$. In the case that u and v are in different components we know that there exist an edge (a path of length one) between them in $\overline{G}$. In the case that u and v are in the same component, say C_i, we can construct a path of two edges between them in $\overline{G}$ as follows. Pick any vertex w from some other component C_j for $j \neq i$ and note that edges $\{u, w\}$ and $\{w, v\}$ are in $\overline{G}$. Thus u, w, v is a path in $\overline{G}$ and hence $\overline{G}$ is connected.

2. A vertex in G is *central* if its greatest distance from any other vertex is as small as possible. This distance is the *radius* of G.

   (a) Prove that for every graph G

   $$\text{rad } G \leq \text{diam } G \leq 2 \text{ rad } G$$

   Solution: Since the diameter is the longest shortest path in the graph, and the radius is just a particular shortest path, we have rad $G \leq$ diam G. Now, since we can always reach any vertex t by going to the center first, then going to t, incurring a cost of at most twice the radius, we have diam $G \leq 2$ rad G.

   (b) Prove that a graph G of radius at most k and maximum degree at most $d \geq 3$ has fewer than $\frac{d}{d-2}(d-1)^k$ vertices.

   Solution: Let z be a central vertex in G, and let D_i denote the set of vertices of G at distance i from z. Then $\cup_{i=0}^{k} D_i$ is all the vertices in the graph. Clearly, $|D_0| = 1$ and $|D_1| \leq d$. For $i \geq 1$ we have $|D_{i+1}| \leq (d-1)|D_i|$, because every vertex in D_{i+1} is a neighbor of a vertex in D_i (why?), and each vertex in D_i has at most $d-1$ neighbors in D_{i+1} (since it has another neighbor in D_{i-1}). Thus $D_{i+1} \leq d(d-1)^i$ for all $i < k$ by induction, giving

   $$|G| \leq 1 + d\sum_{i=0}^{k-1}(d-1)^i = 1 + \frac{d}{d-2}((d-1)^k - 1) < \frac{d}{d-2}(d-1)^k$$

3. A random permutation π of the set $\{1, 2, \ldots, n\}$ can be represented by a directed graph on n vertices with a directed arc (i, π_i), where π_i is the i'th entry in the permutation. Observe that the resulting graph is just a collection of distinct cycles.

   (a) What is the expected length of the cycle containing vertex 1?

Solution: Consider the construction of the directed graph where we start at vertex 1. Each step we select an unmarked vertex at random and move to that vertex. We then mark that vertex before repeating the process. Once this construction marks vertex 1 we have a cycle. Let $|C|$ denote the length of this cycle. Then:

$$\begin{aligned}E(|C|) &= 1 \times \frac{1}{n} \\ &+ 2 \times \frac{n-1}{n}\frac{1}{n-1} \\ &+ \dots \\ &+ n \times \frac{n-1}{n}\frac{n-2}{n-1}\cdots\frac{1}{2}\frac{1}{1} \\ &= \sum_{i=1}^{n} i\frac{1}{n} \\ &= \frac{1}{n}\frac{n(n+1)}{2} \\ &= \frac{1}{2}(n+1)\end{aligned}$$

(b) What is the expected number of cycles?

Solution: Let $f(n)$ denote the expected number of cycles in a graph on n nodes. It is clear that $f(1) = 1$.

Consider $f(n)$ given $f(n-1)$. With probability $1/n$ the new node permutes to itself resulting in an expected number of cycles of $1 + f(n-1)$ and with the remaining probability the new node permutes to a node other than itself, this case then reduces to the $n-1$ case. Hence:

$$f(n) = \frac{1}{n}\Big(1 + f(n-1)\Big) + \frac{n-1}{n}f(n-1) = \frac{1}{n} + f(n-1)$$

It follows recursively that $f(n) = \sum_{i=1}^{n} 1/i$ which is exactly equal to the nth harmonic number $H(n)$.

4. Let $v_1, v_2, \dots, v_n$ be unit vectors in $\mathbb{R}^n$. Prove that there exist $\alpha_1, \alpha_2, \dots, \alpha_n \in \{-1, 1\}$ such that

$$||\alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_n v_n||_2 \leq \sqrt{n}$$

Solution: This can be shown geometrically using similar ideas to Pythagoras' Theorem. Consider the cosine rule:

$$\|\alpha_i v_i + \alpha_j v_j\|_2^2 = \|\alpha_i v_i\|_2^2 + \|\alpha_j v_j\|_2^2 - 2\|\alpha_i v_i\|_2\|\alpha_j v_j\|_2 \cos\theta$$

Where θ is the angle between $\alpha_i v_i$ and $\alpha_j v_j$.

Fix α_i. Then we can choose α_j such that $\theta \leq \pi/2$. Hence $\cos\theta \in [0,1]$. It follows that, given α_i we can choose α_j such that:

$$\|\alpha_i v_i + \alpha_j v_j\|_2^2 \leq \|\alpha_i v_i\|_2^2 + \|\alpha_j v_j\|_2^2$$

Applying this result recursively gives:

$$\begin{aligned}
\|\alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_n v_n\|_2^2 &\leq \|\alpha_1 v_1\|_2^2 + \|\alpha_2 v_2\|_2^2 + \ldots + \|\alpha_n v_n\|_2^2 \\
&= \alpha_1^2\|v_1\|_2^2 + \alpha_2^2\|v_2\|_2^2 + \ldots + \alpha_n^2\|v_n\|_2^2 \\
&= n \\
||\alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_n v_n||_2 &\leq \sqrt{n}
\end{aligned}$$

5. Consider a graph G on $2n$ vertices where every vertex has degree at least n. Prove that G contains a perfect matching.

   Solution: We will prove this in a slightly roundabout way: we first show that G must contain a Hamiltonian path, and then note that a Hamiltonian path contains our desired perfect matching. (A Hamiltonian path is a path which contains every node of the graph.) A direct proof of this is possible, but this proof is shorter and more elegant.

   Consider the longest path $P = (v_1, v_2, \ldots, v_k)$ in G. All neighbors w of v_1 must be elements of P, otherwise the longer path $(w, v_1, \ldots, v_k)$ in G would contradict the definition of P. Similarly, all neighbors of v_k must also be in P. Now since G is simple, we note that all $\geq n$ neighbors of v_1 must be distinct and lie in P, thus we have the bounds $n+1 \leq k \leq 2n$ on the length of P, where in the lower bound we have also accounted for v_1 itself.

   We claim that there exists $j \in \{1, \ldots, k-1\}$ such that v_j and v_{j+1} are neighbors of v_k and v_1 respectively. Suppose for contradiction that this is not the case. Let $S = \{v_i | v_k \sim v_i\}$ be the set of all neighbors of v_k in P. Let $T = \{v_{i-1} | v_1 \sim v_i\}$ be the set of all path vertices immediately preceding the neighbors of v_1 in P. Note that S and T are disjoint by our assumption. Since v_1 and v_k have at least n neighbors in P, we have

   $$|S| + |T| \geq n + n = 2n \geq k.$$

   But we also know that both S and T are subsets of $\{v_1, \ldots, v_{k-1}\}$ so $|S \cup T| = |S| + |T| \leq k-1$, a contradiction. Thus there exists j such that $v_1 \sim v_{j+1}$ and $v_k \sim v_j$.

   Then we have the cycle $C = (v_1, v_2, \ldots, v_j, v_k, v_{k-1}, \ldots, v_{j+1}, v_1)$ in G which contains each vertex of P exactly once. Now we claim that $k = 2n$; in that case C contains our desired Hamiltonian path of G. To see this consider a vertex $x \notin C$. Since G is simple and $|C| = k \geq n+1$, one of the n neighbors of x, call it y must lie in C. But then cutting either one of the edges in C incident to y and including the edge $\{x, y\}$ would result in a path longer than P, contradicting our original longest path assumption. Thus, our cycle C must have length $2n$– it must contain a Hamiltonian path.

Finally, we prove that the Hamiltonian path found above contains a perfect matching. Let the Hamiltonian path P be $(v_1, v_2, \ldots, v_{2n})$. Choose the edges $(v_1, v_2), (v_3, v_4), \ldots (v_{2i-1}, v_{2i}), \ldots,$ (v_{2n-1}, v_{2n}). These edges are all in the Hamiltonian path, and every node in the path is present in exactly one of these edges. As the path contains every vertex in the graph, each node of the graph is the endpoint of exactly one of the edges. Thus, this is a perfect matching in the graph, as desired.

6. Let $G = (V, E)$ be a graph and $w : E \to R^+$ be an assignment of nonnegative weights to its edges. For $u, v \in V$ let $f(u, v)$ denote the weight of a minimum $u - v$ cut in G.

   (a) Let $u, v, w \in V$, and suppose $f(u, v) \leq f(u, w) \leq f(v, w)$. Show that $f(u, v) = f(u, w)$, i.e., the two smaller numbers are equal.

   Solution: Let $c = \min(f(u, w), f(w, v))$. Consider the two ends of the smallest path between u and v. We can route c units of flow from u to w and then from w to v. This means $f(u, v) \geq c = min(f(u, w), f(w, v)) = f(u, w)$, which is only possible if $f(u, v) = f(u, w)$.

   (b) Show that among the $\binom{n}{2}$ values $f(u, v)$, for all pairs $u, v \in V$, there are at most $n - 1$ distinct values.

   Solution: We prove this by induction on the number of nodes. The result is clearly true for a graph with 3 nodes from part a. Assume the result for all graphs G' of size n, and consider a graph G with $n + 1$ nodes. There will be a largest edge, pick one of its two vertices, call it v. Order the edges incident upon v in decreasing order: $f_1, f_2, \ldots, f_n$. So f_1 is the largest edge in G. Note that the f_i are sides of triangles of all whom have one edge in the smaller graph G', where there are only $n - 2$ distinct edges by induction hypothesis. We argue that other than f_1, all the other f_i are equal to some edge in G', thus the number of distinct edges in G can only one larger, with the contribution coming from f_1. This is true because of the decreasing ordering on the f_i's and the triangle property from part a, enforcing each $f_2, \ldots, f_n$ be equal to some edge in G'. Thus the addition of v can only add one new distinct edge weight: f_1, making for at most $n - 1$ distinct weights.

7. Let T be a spanning tree of a graph G with an edge cost function c. We say that T has the *cycle property* if for any edge $e' \notin T$, $c(e') \geq c(e)$ for all e in the cycle generated by adding e' to T. Also, T has the *cut property* if for any edge $e \in T$, $c(e) \leq c(e')$ for all e' in the cut defined by e. Show that the following three statements are equivalent:

   (a) T has the cycle property.

   (b) T has the cut property.

   (c) T is a minimum cost spanning tree.

Remark 1: Note that removing $e \in T$ creates two trees with vertex sets V_1 and V_2. A *cut* defined by $e \in T$ is the set of edges of G with one endpoint in V_1 and the other in V_2 (with the exception of e itself).

Solution: In order to show that (a), (b), and (c) are equivalent, it is enough to show: (a) $\Leftrightarrow$ (c), and (b) $\Leftrightarrow$ (a).

(c) $\Rightarrow$ (a): By contradiction suppose T does not have the cycle property: there exists $e' \notin T$ such that $T \cup \{e'\}$ has a cycle C in which there exists $e \in T$ and $e \in C$ where $c(e) > c(e')$. Let tree T' be the tree obtained by adding e' to T and removing e; T' is a tree with cost strictly less than cost of T which is contradicting with T being an MST.

(a) $\Rightarrow$ (c): By contradiction suppose T is not an MST: let e' be the first edge that was picked by Kruskal's algorithm but does not belong to T. Adding e' to T would create cycle C. Since T has cycle property, $c(e) \leq c(e')$, $e \in C$. Therefore, all $e \in C$, $e \neq e'$ have been visited by the Kruskal's algorithm. We have two cases:

case 1: All $e \in C \backslash \{e'\}$ were picked by the algorithm. In this case the algorithm would not pick e' because it creates a cycle with the existing edges.

case 2: There exists $e^* \in C$, $e^* \neq e'$ such that it was not picked by the algorithm. The reason for not picking an edge is that it would create a cycle with the existing edges. However, since e' was the first edge picked by the algorithm that does to belong to T, this would mean that T has a cycle, which is a contradiction.

(a) $\Rightarrow$ (b): By contradiction suppose T does not have the cut property: there exists $e \in T$ such that there exits edge $e' = (v_1, v_2)$ such that $v_1 \in V_1$ and $v_2 \in V_2$ (V_1 and V_2 are the set of vertices of the two connected components after removing e, see Remark 1.), and $c(e') < c(e)$. Since T is connected graph there exist path P_T between v_1 and v_2 such that all the edges of P_T belong to T. Adding e' to P_T creates a cycle in which there exist $e \in T$ where $c(e) > c(e')$ which is contradicting with T having the cycle property.

(b) $\Rightarrow$ (a): By contradiction suppose T does not have the cycle property: there exists edge e' such that when adding it to T and creating cycle C, there exists $e \in C$, where $c(e) > c(e')$. In T, if we remove e, we have two connected components with vertex sets V_1 and V_2. Let v_1, v_2 be the endpoints of e, where $v_1 \in V_1$ and $v_2 \in V_2$. Since $e \in C$ there exists another path between v_1 and v_2 therefore at least one edge from C belongs to cut (V_1, V_2); since all the edges of C belong to T except $e'$and $T \cap cut(V_1, V_2) = \emptyset$, e' should belong to $cut(V_1, V_2)$. However, $c(e') < c(e)$, which contradicts with T having the cut property.

8. Given a graph $G = (V, E)$, a set of vertices $D \subseteq V$ is called a *dominating set* if every vertex in $V \setminus D$ is adjacent to a vertex in D. Suppose $|V| = n$ and the minimum degree of $G = \delta > 0$. Show that G contains a dominating set of size at most:

$$\frac{n \log(1 + \delta)}{1 + \delta}$$

Solution: Consider a randomly sampled subset of vertices, $X \subseteq V$; each vertex $v \in V$ is in X independently with probability p. We will choose the value of p later. Let $Y \subseteq V$ be the set of vertices in $V \setminus X$ that also have no neighbors in X. So $X \cup Y$ is a dominating set of G. We will show by the probabilistic method that there is some X such that $|X \cup Y| \leq \frac{n(\log(1+\delta)+1)}{1+\delta}$.

First we compute the expected size of $X \cup Y$. Notice that because X and Y are disjoint, $|X \cup Y| = |X| + |Y|$. Let X_v be the indicator variable that $v \in X$. So we have, by linearity of expectation:

$$E[|X|] = \sum_{v \in V} E[X_v] = np$$

Similarly, let Y_v be the indicator variable that $v \in Y$. A vertex is in Y if and only if it and all of its neighbors are not in X, which occurs independently with probability p. Then $E[Y_v] = Pr[Y_v = 1] = (1-p)^{d_v+1}$, where d_v is the degree of v, and we have the following:

$$\begin{aligned} E[|Y|] = \sum_{v \in V} E[Y_v] &= \sum_{v \in V} (1-p)^{d_v+1} \\ &\leq \sum_{v \in V} (1-p)^{\delta+1} = n(1-p)^{\delta+1} \end{aligned}$$

We have used the fact that $d_v \geq \delta$ for all $v \in V$ above. Thus combining the above with the fact that $(1-x) \leq e^{-x}$, we have:

$$\begin{aligned} E[|X \cup Y|] = E[|X|] + E[|Y|] &\leq np + n(1-p)^{\delta+1} \\ &\leq np + ne^{-p(\delta+1)} \end{aligned}$$

For any value of p, the above provides an upper bound on the expected size of our random dominating set. So, for any choice of p, we know that there exists some dominating set with size at most $np + ne^{-p(1+\delta)}$. Now, we want to choose p such that this bound is minimized. This minimization occurs at $p = \frac{\log(1+\delta)}{1+\delta}$ (easy to check it is minimum by taking derivatives). Plugging this value of p into our bound we get:

$$\begin{aligned} E[|X \cup Y|] &\leq \frac{n \log(1+\delta)}{1+\delta} + \frac{n}{1+\delta} \\ &= \frac{n(\log(1+\delta)+1)}{1+\delta} \end{aligned}$$

Thus, at this value of p, there must exist some X, such that the size of the dominating set $|X \cup Y| \leq \frac{n(\log(1+\delta)+1)}{1+\delta}$.

9. Consider the following scenario. Due to large-scale flooding in a region, paramedics have identified a set of n injured people distributed across the region who need to be rushed to hospitals. There are k hospitals in the region, and each of the n people needs to be brought to a hospital that is within a half-hour's driving time of their current location (so different people will have different options for hospitals, depending on where they are right now).

   At the same time, one doesn't want to overload any one of the hospitals by sending too many patients its way. The paramedics are in touch by cell phone, and they want to collectively work out whether they can choose a hospital for each of the injured people in such a way that the load on the hospitals is *balanced*: Each hospital receives at most $\lceil n/k \rceil$ people.

Create a polynomial time algorithm that outputs an assignment of people to hospitals if a valid assignment exists and outputs no otherwise.

Solution: We model the problem as a max-flow problem. Let N be the set of people and K be the set of hospitals. We define the network as follows:
Nodes: $K \cup N \cup s, t$ where s is the source and t is the sink.
Edges:

- There is an edge with capacity one from s to every node in N.
- There is an edge with capacity $\lceil \frac{n}{k} \rceil$ from every node in K to t.
- There is an edge with capacity one from a node in $n \in N$ to a node $k \in K$ if n is within a half-hour distance from k.

If the max-flow in this graph has value n then every person can be assigned to a hospital. This can be done in polynomial time by Ford-Fulkerson; because the capacities are integral, there exists an integral max flow, so the assignment of people to hospitals simply follows this flow. If the max flow of this network is less than n, then no valid assignment exists.