COMPSCI 230: Discrete Mathematics for Computer Science 3/24/2020

Recitation 10: Intro to Graph Theory

Created By: David Fischer; adapted from Erin Taylor, Kevin Sun, Alex Steiger

- **5 minutes:** Begin with review of some quick definitions to help with homeworks – it will be helpful to draw some simple examples as you go through definitions.
  - Tree
    A *tree* is a connected graph with no cycles (circuits).
  - Rooted Tree
    A *rooted tree* is a tree with a single distinguished vertex called the root. We then orient the remaining of the graph such that every node has a single parent, and an arbitrary number of children, where the parent of a node u is the unique node connected to u of lower depth than u (with depth of a node v defined as the number of edges between v and the root). It is easy to verify from the definition of a tree that the parent of u must in fact be unique.
  - Binary Tree
    A *binary tree* is a rooted tree where each parent can have at most 2 children.
  - Full $n - ary$ tree
    A *full n-ary* tree is a tree where each node has either exactly 0 or exactly n children.
- **5 minutes:** Briefly mention that folks should review the lecture videos for real world examples of things like colorings and matchings. In essence, we can conceptualize a graph where each node is a class, and two nodes have an edge between them if the classes share a student. Then, we can "color" the graph such that each color represents a room and time; a valid coloring as defined in the lecture videos represents a valid way of assigning rooms to classes such that no student is expected to be at different rooms at the same time. As well, we can conceptualize a bipartite graph where the nodes in one set of the bipartition correspond to teachers and the nodes in the other set of the bipartition correspond to the set of classes C that must be taught, with an edge between a teacher and a class if that teacher can teach that class. If a C-coloring exists, which can be determined using Hall's theorem, for example, there is a way to make sure each class gets taught.
- **5 minutes:** One application of connectivity is the study of how contagion spreads amongst communities of people. For example, we can conceive of a graph where every individual is a node, and there are edges between nodes if those two individuals come into contact with each other. If a graph has only a single connected component, then a single individual can contaminate the entire graph. However, if the graph is entire disconnected, then a single individual will never contaminate any other individual, theoretically speaking. In real world situations, such as the one we find ourselves in now, there are actually additional vectors for the virus to spread - for example, Covid 19 actually lives 3-4 days on surfaces, so this conceptual graph would actually include all objects with which any individual comes into contact. As well, not all cases of exposure for sure result in a case of the virus being spread.

Often, we model these factors as there being some probability that node u infects node v, which can be dialed and depend on many factors. However, in a general sense, the goal of this quarantine in general is to ensure this conceptual graph has a low level of connectivity, and as many disconnected components as possible.

- **10 minutes to work individually, 5 to go over solution:** Application of Halls Theorem - Consider a traditional deck of playing cards, which contains 52 cards, split into 4 *suits*, and of which each suit is split into 13 *ranks*. Consider placing the cards into a grid of 4 rows and 13 columns. Prove that it is always possible to choose a card from each column such that the resulting set of 13 cards contains one card of each rank.

  Proof. Consider the following construction of a bipartite graph $G = (C \cup R, E)$. Each vertex in C corresponds to a column of the grid, and each vertex in R corresponds to a specific rank (2,3,...,King,Ace). For a specific realization of this problem, we draw an edge between each vertex in C to the rank of each card in that column. We want to show that we can choose a card from each column such that the resulting set of cards contains one card of each rank if and only if there exists a perfect matching in the bipartite graph described above. This is fairly plain to see. On one hand, if we have such a set of drawn cards, we can construct a perfect matching in the graph by selecting the edges corresponding to the cards we chose. On the other hand, if we are given a perfect matching, then we can select the cards in each column corresponding to the edges defined by the perfect matching. Since the perfect matching has each vertex in C incident on exactly one edge, we only draw one card from each column, and since the perfect matching has each vertex in R incident on exactly one edge, the set of cards drawn contains each rank exactly once. Now, we show that such a perfect matching must always exist. As seen in the lecture videos, Hall's theorem states that if $\forall A \subseteq B \,.\, |N(A)| \geq |A|$ where $N(A)$ is the set of all neighbors of vertices in A, then a B perfect matching exists. We will show that the bipartite graph defined above must contain a C perfect matching and a R perfect matching (and so contains a perfect matching generally). Consider the family of sets $S = \{S_i \,.\, i \in [1, 13]\}$ defined by the ranks in column i of the grid. An arbitrary subfamily consists of n sets S_i, and contains the rank of $4n$ cards. Since there are only four cards in each suit (read: of each rank), there must be at least n distinct ranks in this subfamily. One way to see this more clearly is to suppose for contradiction that the subfamily representing the ranks of $4n$ cards contains fewer than n ranks. Note that the maximum number of cards that represent d distinct ranks is $4d$, so if the subfamily had fewer than n distinct ranks, it would have at most $4(n-1) < 4n$ cards, which contradicts the definition of the subfamily. For interests sake we can note visually that set S_i is the set of vertices in R which a particular vertex in C maps to, and S_n is an arbitrary collection of such sets. The logic given shows that any collection of n nodes in C must map to a collection of n distinct nodes in R. Therefore, G must contain a C perfect matching. Similarly, consider a family of sets defined by T_i, which is the set of columns with edges to a particular rank i, and subfamily consisting of n sets T_i. Note that again, there are four suits for each rank, so at most 4 columns can map to a single rank, and the size of the subfamily is at most $4n$. This similarly implies there are at least n distinct columns which map to n distinct ranks. So, G must contain a R perfect matching. Since it contains both a C and R perfect matching, G must contain a perfect matching generally speaking, and so there must be a way to select a card from each column such that the resultant set of cards contains exactly one representative of each rank. □

- Prove that in all trees $m = n - 1$ where $n = |V|$ and $m = |E|$.

  Proof. □

- **7 minutes to work individually, 3 to go over solution:** Prove formally that for any undirected, connected graph $G = (V, E)$, $\sum_{v \in V} d(v) = 2m$ where $m = |E|$ and $d(v)$ is the degree of vertex v.

  Proof. We proceed by induction on the number of edges in the graph. Let a graph have no edges. Then, $m = 0$, and no vertex has any edges incident upon it, and so $d(v) = 0$ for all $v \in V$, so for the base case our claim holds. Now, assume our claim holds for any graph with n edges, and consider a graph with $n + 1$ edges. We want to see that $\sum_{v \in V} d(v) = 2n + 2$. Consider some edge $e = (x, y) \in E$. Consider the graph without this edge, call it $G' = (V, E')$, and let $d'(a) \,.\, a \in V$ be the degree of node a in G'. We know that $|E'| = n$, and the degree of every vertex that is not x or y is the same in G' as in G. So, we have $\sum_{v \in V} d'(v) = 2n = \sum_{v \in (V \setminus \{x,y\})} d(v) + d'(x) + d'(y)$. Then note that e adds one to the degree of both x and y, so that $d'(x) + 1 = d(x)$ and $d'(y) + 1 = d(y)$. Then in the original graph G we have $\sum_{v \in V} d(v) = \sum_{v \in (V \setminus \{x,y\})} d(v) + d'(x) + 1 + d'(y) + 1 = \sum_{v \in (V \setminus \{x,y\})} d'(v) + d'(x) + d'(y) + 2 = 2n + 2$, as desired. □

- **12 minutes to work individually, 8 to go over solution:** Suppose you are given a connected undirected graph $G = (V, E)$ and two coins c_1 and c_2 which are placed on two vertices $a, b \in V$. A move consists of sliding both coins to a neighbor of the vertices they currently lie on. That is, if c_1 and c_2 are currently on vertices u_1 and u_2, we denote by $(u_1, u_2) \to (v_1, v_2)$ a move where $(u1, v1), (u2, v2) \in E$. Note that neither coin can stay in place. The goal is to make a sequence of moves so that both coins lie on the same vertex; such a sequence is a solution. Prove that the only (connected) undirected graphs with no solutions are bipartite graphs where the coins start on vertices in different sets of the bipartition.
  *Hint: First show there is a solution if and only if there is a walk between a and b with an even **number of edges**.*

  We begin by proving the hint.

  Claim 1. *There is a solution if and only if there is a walk between a and b with an even number of edges.*

  Proof. $\leftarrow$: Suppose there is a walk between with an even *number of edges*, $a = v_0$ and $b = v_n$, call it $P = \{v_0, v_1, v_2, ..., v_{n-1}, v_n\}$. Note that since there is an even number of edges in the walk, there are an odd number of vertices, and n is even. Then the sequence of moves defined by $\{(v_{i-1}, v_{n-i+1}) \to (v_i, v_{n-i}) \,.\, i \in [1, n/2]\}$ is a valid solution, as can be seen by plugging in $i = n/2$: $\{(v_{n/2-1}, v_{n/2+1}) \to (v_{n/2}, v_{n/2})\}$ is a move which results in both coins resting on the same vertex.
  $\to$: Suppose there is a solution, so the following sequence of moves of length k exists: $\{(a, b) \to (u_1, v_1), (u_1, v_1) \to (u_2, v_2), ..., (u_{k-1}, v_{k-1}) \to (u_k, v_k)\}$, where $u_k = v_k$. It follows that the walk $\{a, u_1, u_2, ..., u_{k-1}, u_k, v_{k-1}, v_{k-2}, ..., v_1, b\}$ is a walk from a to b with length of $2k$, which must be even. □

  It will be useful to have following.

  Claim 2. *Any walk between nodes on the same side of the bipartition has even length, and any walk between nodes on different sides of the bipartition has odd length.*

  Proof. We proceed by induction on the number of nodes. Base case is when the number of vertices $n = 2$, so the vertices are $\{u, v\}$, the set of edges is $\{(u, v)\}$ with $A = u$ and $B = v$. Then any walk from bipartite set A to B

starts $\{u, v\}$, and proceeds by adding an arbitrary number of copies of $\{v, u, v\}$. Clearly, this walk must have odd length. A similar logic follows showing that walks to and from the same bipartite set have even length. Assume our claim holds for a graph with n vertices, and consider a bipartite graph with $n+1$ vertices and bipartite sets A and B. Remove some vertex v from set A such that the result graph is still connected, and note that the length of all walks from nodes in set B to nodes in set A in the new graph are odd, and vice versa, by the inductive hypothesis. As well, the length of all walks from A to A (similar for B) have even length. Consider all neighbors of the node we removed, call some such neighbor u. Note that since $v \in A$, and G is bipartite, we have that $u \in B$. Again by the IH, we have that the length of any walk from a node in B to u is even. Then, following the edge (u, v) makes the walk length odd. Similarly, the length of any walk from a node in A to u is odd, and adding the edge (u, v) makes the walk even. So, the claim is shown for all graphs with no fewer than 2 vertices. □

Now, we show the theorem.

Theorem 1. *A connected undirected graph has no solutions if and only if the graph is bipartite and the coins start on vertices in different sets of the bipartition.*

Proof. $\rightarrow$ We must show that if a graph G is connected and undirected with no solutions, then G is bipartite and a and b are in different sets of the bipartition. It suffices to show that if G is not bipartite or G is bipartite and a and b are in the same set of the bipartition then there is a solution. By Claim 1 we want to show there is an even walk between a and b. So, we proceed by cases on whether G is bipartite. Suppose G is not bipartite. From lecture, we know then that G contains an odd length cycle, C. Now, consider a path between a and some other node in the odd length cycle $y \in C$ and call it $P_{a,y}$, and similar for b to y with $P_{b,y}$. If $|P_{a,y}| + |P_{b,y}|$ is even, then we have found an even walk between a and b, and so there is a solution. If $|P_{a,y}| + |P_{b,y}|$ is odd, then consider the odd length walk $P_{y,y}$ from y to y found by following the edges of the cycle C. Now note that following the edges in $P_{a,y}$, then $P_{y,y}$ and finally $P_{b,y}$ results in an even length walk, and so a solution must exist. On the other hand, suppose G is bipartite and a and b are in the same set of the bipartition. Then, there is an even walk between a and b by Claim 2, and so there is a solution.
$\leftarrow$ For this direction, we want to show that if the graph is bipartite and the coins start on vertices in different sets of the bipartition, then the graph has no solutions. It suffices to show (again by contrapositive) that if the graph has a solution then, the graph is not bipartite or the coins start on vertices in the same set of the bipartition. We only need consider the case where the graph is bipartite. Since there is a solution, there is an even length walk between a and b, which as shown in Claim 2 only occurs when a and b are vertices in the same set of the bipartition. Therefore, we have shown both directions, and a connected, undirected graph has no solutions if and only if the graph is bipartite and the coins start on vertices in different sets of the bipartition. □

- **10 minutes to work individually, 5 to go over solution:** Consider the *tournament* graph $G = (V, E)$, which is a directed graph where exactly one of (a, b) and (b, a) appear in the edge set for each pair of distinct vertices $\{a, b\}$. A *champion* of the graph is a vertex which has a path to every other vertex of the graph of length at most two. In other words, every vertex besides the champion is either a neighbor of the champion graph, or a neighbor of a neighbor of the champion. Prove that a vertex of G with largest out degree is a champion vertex.

Proof. Suppose for contradiction that a node u with maximal degree is not a champion. Let X be the set of out neighbors of u. Since u is not a champion, there is a vertex v that is not a neighbor of u, and which is not a neighbor of any neighbor of u. In other words, $\forall x \in X \cup \{u\} | (x, v) \notin E$. Now, since G is a tournament graph, we have that since $(x, v) \notin E$, we must have $(v, x) \in E$. However, this means that the out degree of v is $|X| + 1$ (since it must connect to everything in $|X|$, and to u). This contradicts the fact that u has maximal out degree. So, u must in fact be a champion, and any node with maximal degree is a champion (as a result of this, all tournament graphs must have at least one champion). $\square$