CS 30: Discrete Math in CS (Winter 2019): Lecture 26

Date: 25th February, 2019 (Monday)

Topic: Graphs: Handshake Lemma, Walk and Paths

Disclaimer: These notes have not gone through scrutiny and in all probability contain errors.
Please discuss in Piazza/email errors to deeparnab@dartmouth.edu

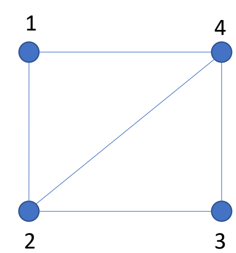
---

1. **What is a graph?**

   A graph $G = (V, E)$ is defined as a pair of sets (which will be finite sets for this course). The first set is V called the set of *vertices* or set of *nodes*. The second set E is called the set of *edges* or set of *links*. Elements of E are *unordered pairs* (subsets of size 2) of distinct vertices.

   Example: Say $V = \{1, 2, 3, 4\}$. Say $E = \{(1, 2), (2, 3), (1, 4), (3, 4), (2, 4)\}$. Then $G = (V, E)$ is one graph on 4 vertices and 5 edges.

   Pictorial Representation. Almost everyone I know thinks about graphs *pictorially*. The vertices of the graph are drawn as points/circles on the plane. The edges are drawn as straight (or sometime non-straight) lines. For example, the graph above is pictorially represented as follows



   Remark: *The pictures are good for intuition. The final proof however, as you know, should never be using a picture. It should be formal, and I'll try to give proofs with only words. A picture is okay for illustration, not demonstration.*

2. **Why graphs?** Graphs are amazing objects to argue about objects which have pairwise relations between them. Perhaps the graph which affects all out lives is the **Web Graph**. The nodes are all web-pages in the world; there is an edge between two web-pages if they link each other. Then there is the **Social/Facebook Graph**. The nodes are individuals; there is an edge between two nodes if they are friends.

   But graphs come up every where. Molecules are often modeled as graphs in computational biology. Agents and Items they wish to purchase are often modeled as graphs in economics. Processors and Jobs are modeled as graphs in scheduling. The list is endless, and **Graph Theory** is an extremely important object of study.

3. **Notations.** There is a lot of notation in graph theory; but they are often picturesque and intuitive. One of the goals of this module is to actually acquaint you of these. Below we fix a graph $G = (V, E)$.

   - Given an edge $e = (u, v)$, the vertices u and v are the ***endpoints*** of e. We say e ***connects*** u and v. We say that u and v are ***incident*** to e.
   - Two vertices $u, v \in V$ are ***adjacent*** or ***neighbors*** if and only if (u, v) is an edge.
   - The ***incident edges*** on v is denoted using the set $\partial(v)$. So,

     $$\partial_G(v) := \{(u, v) : (u, v) \in E\}$$

     We lose the subscript if the graph G is clear from context.
   - Given a vertex v, the ***neighborhood*** of v is the set of neighbors of v. This is denoted sometimes as $N(v)$ or sometimes as $\Gamma(v)$. So,

     $$N_G(v) := |\{(u, v) : (u, v) \in E\}|$$

     if the graph G is clear from context.
   - The cardinality of $N_G(v)$ is called the ***degree*** of vertex v. We denote it using $\deg_G(v)$. This counts the number of neighbors of v. Note that,

     $$\deg_G(v) = |N_G(v)| = |\partial_G(v)|$$

   - A vertex v is ***isolated*** if its degree is 0. That is, it has no edges connected to it.
   - A graph $G = (V, E)$ is called ***regular*** if all degrees are equal, that is, $\deg_G(v) = \deg_G(u)$ for all u and v.
   - Given a graph $G = (V, E)$, we use $V(G)$ to denote V and $E(G)$ to denote E. This notation is useful when we are talking about multiple graphs.

4. **Deleting and Inserting Edges and Vertices from a graph.**

   Fix a graph $G = (V, E)$. Let $e = (u, v)$ be an edge in E. We get a new graph by ***deleting*** the edge e from G. This graph is denoted as $G - e$ or $G \setminus e$. $V(G \setminus e) = V$ and $E(G \setminus e) = E \setminus e$.

   $$G - e := G \setminus e := (V, E \setminus e)$$

   Note $|V(G \setminus e)| = |V(G)|$ but $|E(G \setminus e)| = |E(G)| - 1$.

   Given a susbet $F \subseteq E$ of edges, we can delete all the edges in F iteratively to get the graph $G - F$ (this is not a usual notation). In particular, we get the graph G' defined as $G' = (V(G), E(G) \setminus F)$.

   Similarly, we can *add/insert* edges to G. Let $e = (u, v)$ be a pair of vertices. Then, we get a new graph by *inserting* the edge e in G. This graph is denoted as $G + e$ or $G \cup e$. Note if e was already present in E, then $G + e = G$.

   $$G + e := G \cup e := (V(G), E(G) \cup e)$$

We can also delete a *vertex*. When we delete a vertex, we delete that vertex from the vertex set and *also* all the edges adjacent to v. This new graph is called $G - v$ or $G \setminus v$.

$$G - v := G \setminus v = (V(G) \setminus v,\ E(G) \setminus \partial(v))$$

Note that, and this is going to be useful, $|V(G-v)| = |V(G)|-1$ AND $|E(G-v)| = |E(G)| - \deg_G(v)$. Note that $|E(G-v)|$ may be equal to $|E(G)|$; this occurs if v was an isolated vertex in G.

5. **Subgraphs and Induced Subgraphs.**

   A graph $H = (W, F)$ is a ***subgraph*** of a graph $G = (V, E)$ if $V \subseteq W$ and $F \subseteq E$, and if (W, F) is a valid graph. That is, for any edge $(u, v) \in F$, both u and v are in the set W

   Given a graph $G = (V, E)$ and a subset $W \subseteq V$ of vertices, the ***induced subgraph*** $G[W] = (W, F)$ where $F \subseteq E$ and *any* original edge $(u, v) \in E$ with both endpoints $u, v \in W$ lies in F.

6. **The Handshake Lemma.** The first proof in graph theory is something you have already seen before in a UGP.

Theorem 1. For any graph $G = (V, E)$, we have

$$\sum_{v \in V(G)} \deg_G(v) = 2|E(G)| \tag{1}$$

Proof. We have seen a "counting two ways" proof of it in a previous UGP. I am going to utilize this opportunity to give a *different* proof based on induction. Induction on graphs is something we need to get used to. Fast.

Let $P(n)$ be the predicate which is true if *for all* graphs $G = (V, E)$ with $|V| = n$, (1) holds. The theorem is proved if we show $\forall n \in \mathbb{N} : P(n)$ is true. This predicate should remind you of predicates we defined for proving correctness of code.

Base Case: Is $P(1)$ true? We need to show for all graphs $G = (V, E)$ with $|V| = 1$, (1) holds. If $V(G) = \{u\}$, then note that $E(G) = \emptyset$ (there are no vertices to connect). Therefore, u is an isolated vertex. $\deg(u) = 0$. Thus, the LHS of (1) is 0 and so is the RHS.

Inductive Case: Fix a natural number $k \geq 1$. Assume $P(k)$ is true. We need to show $P(k+1)$ is true. That is, we need to show *for any* graph with $k+1$ vertices, (1) holds. To that end, fix a graph $G = (V, E)$ with $|V| = k + 1$.

Let w be an *arbitrary* vertex in G. Consider the graph $G' := G - w$ (recall the definition of deletion of a vertex from above). Now observe the following:

$$\text{For all } u \in N_G(w), \quad \deg_{G'}(u) = \deg_G(u) - 1 \tag{2}$$

$$\text{For all } u \in V(G') \setminus N_G(w), \quad \deg_{G'}(u) = \deg_G(u) \tag{3}$$

That is, in the new graph G', the degrees of (in G) neighbors of w go down by 1, while the degrees of other nodes remain unchanged. Also note by Induction Hypothesis,

$$\sum_{v \in V(G')} \deg_{G'}(v) = 2|E(G')| \tag{IH}$$

This is because $|V(G')| = |V(G)| - 1 = k$.

Finally, note

$$|E(G')| = |E(G)| - |\partial_G(w)| = |E(G)| - |N_G(w)| \tag{4}$$

Now we have all the ingredients to complete the proof. We get

$$\begin{aligned}
\sum_{v \in V(G)} \deg_G(v) &= \deg(w) + \sum_{v \in V(G) \setminus w} \deg_G(v) && \text{Pulling } w \text{ out.} \\
&= |N_G(w)| + \sum_{v \in V(G')} \deg_G(v) && \text{Definition of degree.} \\
&= |N_G(w)| + \sum_{v \in N_G(w)} \deg_G(v) + \sum_{v \in V(G') \setminus N_G(w)} \deg_G(v) \\
&= |N_G(w)| + \sum_{v \in N_G(w)} (\deg_{G'}(v) + 1) + \sum_{v \in V(G') \setminus N_G(w)} \deg_{G'}(v) && \text{Using (2), (3)} \\
&= 2|N_G(w)| + \sum_{v \in V(G')} \deg_{G'}(v) && \text{Collecting Terms} \\
&= 2\left(|N_G(w)| + |E(G')|\right) \\
&= 2|E(G)| && \text{Using (4)}
\end{aligned}$$

We established $P(k+1)$ and thus, by induction, $P(n)$ is true for all n. That is, the theorem holds. □

7. **Perambulations in Graphs.** We introduce a lot of definitions involving alternating sequence of vertices and edges. These are key definitions so make sure you understand them. Throughout below we fix a graph $G = (V, E)$.

   - A ***walk*** w in G is an *alternating sequence* of vertices and edges

     $$w = (v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k)$$

     such that the ith edge $e_i = (v_{i-1}, v_i)$ for $1 \le i \le k$. Intuitively, imagine starting at vertex v_0, using the edge e_1 to go to the adjacent vertex v_1, and then using e_2 to go to the adjacent (to v_1) vertex v_2, and so on and so forth till we reach v_k. Note, by this constraint above the identity of the edges are defined by the vertices, and so telling them explicitly is redundant. Nevertheless, when talking about a walk, one explicitly writes down the edges.

     Note both the edges and vertices could repeat themselves. That is e_i could be the same as e_j for $j \ne i$. In fact, e_{i+1} could be the same as e_i; this would mean going from one endpoint of e_i to the other and immediately returning back.

The walk above is said to *start* at v_0 and *end* at v_k. The node v_0 is often called the *source/origin* and the node v_k is often called the *sink/destination*. If there is a walk as described above, then we often say "there is a walk from v_0 to v_k."

A walk is of ***length*** k if there are k edges in the sequence. Note that since repetition of both vertices and edges are allowed, walks could go on for ever.

- A ***trail*** t in G is a walk with no edges repeating. That is, a trail is also an alternating sequence of vertices and edges

  $$t = (v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k) \quad \text{where the } e_i\text{'s are distinct}$$

  Note that a trail *could repeat vertices*. For instance, if the graph was $G = (\{a, b, c, d, e\}, \{(a, b), (b, c), (c, d), (d, b), (b, e)\})$, then the following is a valid trail. The vertex b is repeated.

  $$t = (a,\ (a, b),\ b,\ (b, c),\ c,\ (c, d),\ d,\ (d, b),\ b,\ (b, e),\ e)$$

  Also note that a trail cannot be arbitrarily long. A trail's length is at most $|E|$.

- A ***path*** p in a graph G is a walk with no *vertices repeated*. Note that a path is always a trail. In fact, a path is a trail with no vertices repeating. Oftentimes, for describing paths, the alternating edges are dropped. So for instance

  $$p = (v_0, v_1, \ldots, v_k) \text{ actually stands for } (v_0,\ (v_0, v_1),\ v_1,\ (v_1, v_2),\ v_2, \cdots, (v_{k-1}, v_k),\ v_k)$$

Theorem 2. Let $G = (V, E)$ be a graph and u and v be two distinct vertices in $V(G)$. If there is a *walk* from u to v in G, then there is a *path* from u to v.