CS 30: Discrete Math in CS (Winter 2019): Lecture 27

Date: 27th February, 2019 (Wednesday)
Topic: Graphs: Connectivity, Trees
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. **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 $1 \leq i \leq k$. 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 $j \neq i$. 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

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

- A ***closed walk*** is a walk whose origin and destination are the same vertex. If $e = (u, v)$ is an edge in G, then the following is a closed walk of length 2

$$w = (u, e, v, e, u)$$

  A closed walk must be of length at least 2.

  Note that given a closed walk, we can choose any $v_i \in w$ to be the source *and* the destination using the same vertices and eges of the closed walk. That is, given a closed walk

$$w = (v_0, e_1, v_1, e_2, v_2, \dots, e_k, v_k) \quad \text{with } v_k = v_0$$

  and an arbitrary vertex $v_i \in w$ with $1 \le i < k$, we can have another closed walk

$$w' = (v_i, e_{i+1}, v_{i+1}, \dots, e_k, v_k = v_0, e_1, v_1, e_2, v_2, \dots, e_i, v_i)$$

  Note w' is a closed walk whose source and destination are v_i.

- A ***circuit*** is a closed trail of length at least 1. That is, it is a trail whose origin and destination are the same vertex, and contains at least one edge. The latter constraint disallows a singleton node from being defined as a circuit. Indeed, a circuit must have at least 3 edges – do you see this?
- A ***cycle*** is a circuit with no vertex other than the source and destination repeating. Thus, a cycle is a path followed by an edge from the destination of the path to the origin, and then the origin node.

Theorem 1. Let $G = (V, E)$ be a graph and u and v be two distinct vertices in $V(G)$. If there is a *finite walk* from u to v in G, then there is a *path* from u to v.

Proof. In the UGP, you see a way to prove the above by induction. There is another (slicker) way of looking at the inductive proof. It involves the "minimal counter example" idea. Goes like this.

Let W be the set of all walks from u to v. We know that there is one of finite length. Pick $w \in W$ to be the walk from u to v of the ***smallest length***. We claim that this walk must be a path.

Suppose not. Suppose w is not a path. That is,

$$w = (x_0 := u, e_1, x_1, \dots, e_k, x_k := v)$$

but two vertices, say x_i and x_j with $i < j$ and both $0 \le i, j \le k$, are the same. Then, consider the walk

$$w' = (x_0 := u, e_1, x_1, \dots, x_i, e_{j+1}, x_{j+1}, \dots, e_k, x_k)$$

This walk w' is a smaller length walk than w. But this contradicts the choice of w. Thus, our supposition must be wrong. Therefore, w is a path.

□

Theorem 2. Let $G = (V, E)$ and u be an arbitrary vertex in $V(G)$ and e be an arbitrary vertex in $E(G)$. If there is a *circuit* in G containing u, then there is a *cycle* in G containing u. If there is a *circuit* in G containing e, then there is a *cycle* in G containing e.

2. **Connectivity in Graphs**

   In a graph $G = (V, E)$, we say a vertex v is ***reachable*** from u if there is a path starting from v and ending at u.

   A graph $G = (V, E)$ is ***connected*** if any vertex u is reachable from another vertex v. A graph is ***disconnected*** otherwise.

   Given any graph $G = (V, E)$, we can partition it into ***connected components***. That is, $V = V_1 \cup V_2 \cup \cdots \cup V_k$ where (a) any two vertices in the same V_i are reachable from one another, and (b) a vertex $u \in V_i$ is *not* reachable from any vertex $v \in V_j$ if $i \neq j$.

   Given any graph $G = (V, E)$ and a vertex $u \in V$, the set of vertices $S_u \subseteq V$ which are *reachable* from u is the connected componenet of G which contains u.

3. **Trees.**

   A graph $G = (V, E)$ is a ***forest*** if it doesn't contain any cycles.

   A forest $G = (V, E)$ is a ***tree*** if it is connected. That is, a tree $G = (V, E)$ is a connected graph which doesn't contain any cycles.

Theorem 3. Let $G = (V, E)$ be a forest. Then each connected component of G induces a tree.

Proof. Let $V_1, \ldots, V_k$ be the connected components of G. Each $G[V_i]$ is connected by definition. If G doesn't contain a cycle, then any subgraph also doesn't contain a cycle. Thus, $G[V_i]$ contains no cycle. Thus $G[V_i]$ is a tree. □

There are many equivalent ways to think about trees. We prove some here, and some are left as exercises in the UGP.

Theorem 4. (The Tree Theorem.) Let $G = (V, E)$ be a graph. The following are equivalent statements.

(a) G is a tree.

(b) G has no cycles and adding any edge to G creates a cycle.

(c) Between any two vertices in G there is a unique path.

(d) G is connected, and deleting any edge from G disconnects the graph, and the resulting graph has exactly two connected components.

(e) G is connected and $|E| = |V| - 1$.

(f) G has no cycles and $|E| = |V| - 1$.

Proof.

$(a) \Rightarrow (b)$: Since G is a tree, G doesn't contain any cycle. Suppose we add an edge $(u, v) \notin E$ to G. Note, there is a path from v to u in G since G is connected. Then in the graph $H := G + (u, v)$, there is a cycle which starts from v, follows the path to u, and then returns to v using the new edge (u, v).

$(b) \Rightarrow (c)$: Fix any two vertices u and v in G. Since G has no cycles, there cannot be *two or more than two* paths in G from u to v. (This is established in the UGP). We now show it has one path from u to v. If the $(u, v) \in E$, this is the length 1 path. Otherwise, since $G + (u, v)$ forms a cycle and G had no cycles, this cycle must contain the edge (u, v). Deleting (u, v) from this cycle leads to a path from u to v. More precisely, given the cycle containing (u, v), I could write it as a closed walk where the starting and ending vertex is u. Either the *second* or the *penultimate* vertex must be v. Deleting (u, v) would give a walk from v to u or from u to v. In either case, we would have a path from u to v.

$(c) \Rightarrow (d)$. UGP

$(d) \Rightarrow (e)$. We prove this by induction. Let $P(n)$ be true if "for all graphs $G = (V, E)$ with $|V| = n$ which are (a) connected, and (b) deleting any edge e from G disconnects G and leads to exactly two connected components, must have $|E| = |V| - 1$."

Base Case: Is $P(1)$ true? There is only graph with 1 vertex. It has $|E| = 0 = 1 - 1$. Thus, $P(1)$ is vacuously true.

Inductive Case: Fix a natural number k and assume $P(1), P(2), \ldots, P(k)$ is true. We wish to prove $P(k+1)$ is true. To this end, we fix a graph $G = (V, E)$ with $|V| = k+1$ which satisfies the condition (d). Fix any edge $(u, v) \in G$. (d) implies that $G - (u, v)$ has two connected components V_1 and V_2. Consider the two induced subgraphs of G, namely, $G_1 = G[V_1]$ and $G_2 = G[V_2]$. Note that for both these graphs (a) they are connected, and (b) deleting any edge disconnects the graph into two connected components. Furthermore, $|V_1|$ and $|V_2|$ are both $\leq k$. Thus, by strong induction, $|E(G_1)| = |V_1| - 1$ and $|E(G_2)| = |V_2| - 1$. Thus, $|E(G)| = |E(G_1)| + |E(G_2)| + 1 = |V_1| + |V_2| - 1 = |V| - 1$.

$(e) \Rightarrow (f)$: Before we prove this, we make an important observation. Since $\sum_{v \in V} \deg(v) = 2|E|$ and since $|E| = |V| - 1$, we get that there exist some $v \in V$ with $\deg(v) < 2$. If every $\deg(v) \geq 2$, then the LHS is $\geq 2|V|$. Furthermore, since G is connected, there can't be any isolated vertices. Thus, $\deg(v) \geq 1$ for all v. Together, we can conclude *there must exist a vertex in G with* $\deg(v) = 1$. *This is called a* ***leaf*** *of the tree.*

We now prove $G = (V, E)$ has no cycles by induction. I am going to sketch the proof; I expect by now you to be comfortable enough to write the whole formal proof using predicates and all.

Since $G = (V, E)$ has a leaf, let's call it v. Consider $G' = G - v$. Note, $|E(G')| = |E(G)| - 1 = |V(G)| - 2 = |V(G')| - 1$. Furthermore, G' is connected. For any $x, y \in V(G')$, there was a path from x to y in G. This path couldn't have contained v since $\deg_G(v) = 1$. Thus, this path also exists in G', implying G' is connected. Thus, by induction, G' doesn't have any cycles. This means G doesn't have any cycles since $\deg_G(v) = 1$ implies v can't introduce new cycles.

$(f) \Rightarrow (a)$: UGP $\square$

Theorem 5. Let $G = (V, E)$ be a forest with k connected components. Then $|E(G)| = |V(G)| - k$.

Proof. Let $V_1, \ldots, V_k$ be the connected components of $G = (V, E)$. We know that each $G_i := G[V_i]$ is a tree. Thus, $|E(G_i)| = |V(G_i)| - 1$. Thus, $|E(G)| = \sum_{i=1}^{k} |E(G_i)| = \left(\sum_{i=1}^{k} |V(G_i)|\right) - k = |V(G)| - k$. $\square$