Introduction to graphs

Degrees

Read Sections 1,2,3,5,6 from Chapter 1. These sections contain a lot of definitions that we will not imemediately use, but you will at least remember where to find them when they come up later. A graph is a pair $G(V, E)$, where V is a set of *vertices* (nodes) and $E \subseteq \binom{V}{2}$ a set of *edges* (links). In this course, we will almost always consider graphs to be *simple*, meaning that they have no self-loops or multiple edges, and that edges are undirected. Let $d(v)$ indicate the *degree* of a vertex $v \in V(G)$, i.e. the number of edges that are *indicent* to it (contain v), or the number of vertices *adjacent* to v (that share an edge with v). As a warm-up exercise, we prove the Handshake Lemma.

Lemma 1 (Handshake Lemma.)**.** *For any graph G, we have* p.5

$$2|E| = \sum_{v \in V} d(v).$$

It follows immediately that the number of odd-degree vertices in a graph is always even.

Exercise 1. *Is it possible to have a graph G (on at least 2 vertices) such that $d(v) \neq d(u)$ for all $v \neq u \in V(G)$?* 1 point

Solution. When we consider simple graphs, the set of possible vertex degrees is $\{0, 1, 2, \ldots, n-1\}$. (Why?) This means that if we have n vertices, they must use exactly all of those degrees in order to avoid repetition. However, if there is a vertex of degree 0, that means it has an edge to **no** other vertices. A vertex of degree $n-1$ has an edge to **all** other vertices. This is a contradiction, since these two vertices cannot share an edge and a non-edge at the same time. Therefore, there is no such graph.
We denote the average degree of G as $d(G)$, the minimum as $\delta(G)$ and the maximum as $\Delta(G)$. A *subgraph* of G is a graph H such that $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. Note that we cannot take any subset of $V(G)$ and $E(G)$, as this is not necessarily a graph. Of course a graph can have a large average degree and small minimum degree. However, if a graph has high average degree it must at least have a subgraph with a high minimum degree. We show the following Proposition.

Proposition 2. *Every graph G with at least one edge has a subgraph H with $\delta(H) \geq d(G)/2$.* 1.2.2, p.6

Paths and cycles

A *path* is a graph of the form $P(V, E)$ with $V = \{v_0, \ldots, v_k\}$ and $E = \{v_0v_1, \ldots, v_{k-1}v_k\}$. A *cycle* is a graph of the form $C(V, E)$ with $V = \{v_0, \ldots, v_k\}$ and $E = \{v_0v_1, \ldots, v_{k-1}v_k, v_1v_k\}$. We say that a graph is *connected* if there exists a path in the graph between any pair of vertices. In class, we will prove the following Proposition.

Proposition 3. *Every graph G contains a path of length at least $\delta(G)$ and a cycle of length at least $\delta(G) + 1$.* 1.3.1, p.8

A *walk* in a graph is a generalization of a path: it may repeat vertices and edges. A *closed walk* in a graph is a generalization of a cycle: it may repeat vertices and edges.

Exercise 2. *Show that if a simple graph G contains an odd closed walk, it contains an odd cycle.* 2 points

Solution. We consider walks of length ≥ 3, since a walk of length 1 cannot be closed in a simple graph. It is not so hard to verify that a closed walk of length 3 must be a cycle. We proceed by induction. Suppose that every odd closed walk of length $< k$ contains an odd cycle, for an odd number $k > 3$, and suppose we have an odd walk of length k: $v_0, v_1, \ldots, v_k = v_0$. If there are no repeated vertices other than $v_0 = v_k$, then this odd walk is an odd cycle and we are done. Otherwise, suppose that $v_i = v_j$ for $0 \leq i < j < k$. Then the sequence $v_i, v_{i+1}, \ldots, v_j$ forms a closed walk, and the sequence $v_j, v_{j+1 \pmod k}, \ldots, v_i$ forms a closed walk as well. The lengths of these two closed walks add up to k (and both of them are $< k$) and therefore one of them is an odd walk of length $< k$ which, by the inductive hypothesis, contains an odd cycle.
A *trail* in a graph is a generalization of a path: it may repeat vertices, but not edges. A *tour* in a graph is a generalization of a cycle: it may repeat vertices, but not edges. An *Euler tour* in a graph is a tour that visits every edge. One of the earliest and most famous theorems in graph theory is due to Euler. It is inspired by the Seven Bridges of Königsberg problem.

Proposition 4. *A connected graph has an Euler tour if and only if every vertex has even degree.* 1.8.1, p.22

Trees

A *tree* is an acyclic connected graph. The following Theorem summarizes a set of equivalent definitions of trees, which are each useful in different contexts.

Theorem 5. *The following are equivalent for a graph T:* 1.5.1, p.14

(i) T is a tree;

(ii) any two vertices are linked by a unique *path in T;*

(iii) T is minimally connected ($T - e$ is disconnected for every $e \in E(T)$);

(iv) T is maximally acyclic ($T + e$ has a cycle for every $e \in \overline{E(T)}$).

Exercise 3. *Prove Theorem 5.* 3 points

Solution. We will not prove the entire Theorem here, but we will focus on what is likely the hardest part of the proof.

Claim 6. *If a graph G with two vertices $v, w \in V(G)$ contains two distinct vw-paths P_1 and P_2, then G contains a cycle.*

Proof. Label the two paths as $P_1 : \; v = x_0, x_1, \ldots, x_k = w$ and $P_2 : \; v = y_0, y_1, \ldots, y_l = w$. Let t be the smallest i such that $x_i \neq y_i$ (the first time the paths diverge). Such an i must exist since the paths diverge at some point, and have the same endpoints (so one cannot be a subpath of the other). Then, let s be the smallest number that is greater than t such that x_s is on both paths P_1 and P_2. Then, $x_s = y_{s'}$ for some $t < s' \leq l$. Then, consider the two subpaths: $P_1' : \; x_{t-1}, \ldots, x_s$ and $P_2' : \; y_{t-1}, \ldots, y_{s'}$. By construction, these two paths have the same endpoints, but do not share any inner vertices. Therefore, together they must form a cycle. □

Exercise 4. *Prove that every tree has at least $\Delta(T)$ vertices of degree 1 (leaves), where Δ indicates the maximum degree of T.* 1 points

Solution. We proceed by induction on $|V(T)|$. It is easy to check that this claim holds for $|V(T)| \in \{0, 1\}$. Let T be a tree and $v \in V(T)$ a vertex of maximum degree $d(v) = \Delta(T)$. If we delete v from T, we claim that this leaves $\Delta(T)$ connected components in $T - v$. Every pair of neigbors w_1, w_2 of v has a unique $w_1 w_2$-path through v, and therefore must lie on different components in $T - v$. Furthermore, every connected component of $T - v$ must contain a neighbor w_i of v. This holds because every vertex x in $V(T) \setminus v$ has a xv-path in T on which the penultimate vertex is a neighbor of v. Therefore, every vertex $x \in V(T) \setminus v$ has a path to some neighbor of v that does not contain v itself. For every component C_i of $T - v$ with neighbor w_i, we have three cases. Suppose w_i is an isolated vertex,, Then it was a leaf in T. Suppose that $C_i \sim K_2$, then it has 2 leaves. Otherwise, C_i satisfies the inductive hypothesis and has at least 2 leaves. One of those 2 leaves may be w_i, but the other one must then be a leaf in the original tree T. Therefore, we find one leaf for each C_i.

Note the following Corollary.

Corollary 7. *Every tree on at least two vertices has at least two leaves.*

Exercise 5. *Show that a tree without a vertex of degree 2 has more leaves than inner vertices. Can you find a very short proof of this?* 1 points

Bipartite graphs

A *bipartite graph* is a graph $G(V, E)$ if there exists a bipartition $V = V_1 \cup V_2$ (with $V_1 \cap V_2 = \emptyset$) such that every edge has one endpoint in V_1 and one endpoint in V_2. We discuss the following equivalent definition in class.

Proposition 8. *A graph is bipartite if and only if it contains no odd cycles.* 1.6.1, p.18

Exercise 6. *Let $V = \{0, 1\}^d$. In other words, V is the set of all binary strings of length d. The d-dimensional hypercube Q_d is the graph on V such that two vertices in V share an edge if and only if the strings differ in exactly one bit. Show that the hypercube Q_d is a bipartite graph, for $d = 1, 2, \ldots$* 1 point

Solution. Note that if two binary strings differ in exactly one bit, then the parity of their sums must differ. Therefore, if we partition the strings by sum parity, we obtain a bipartition in the hypercube graph.

Exercise 7. *Show that if a bipartite graph G is k-regular, meaning that $d(v) = k \ \forall v \in V(G)$, then the partition classes have the same size.* 1 point

Solution. Let $V = A \cup B$ be the bipartition of $V(G)$. Note that all edges have exactly one endpoint in A. Therefore $|E| = \sum_{a \in A} d(a) = k \cdot |A|$. However, this is also true for B, which gives us $|E| = \sum_{b \in B} d(a) = k \cdot |B|$, and therefore we must have $|A| = |B|$.