Introduction

Lecture 1 – Graph Theory 2016 – EPFL – Frank de Zeeuw

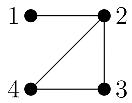
1 Definitions

Definition. *A* graph $G = (V, E)$ *consists of a finite set* V *and a set* E *of two-element subsets of* V. *The elements of* V *are called* vertices *and the elements of* E *are called* edges.

For instance, very formally we can introduce a graph like this:

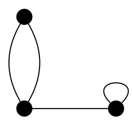
$$V = \{1, 2, 3, 4\}, \quad E = \{\{1, 2\}, \{3, 4\}, \{2, 3\}, \{2, 4\}\}.$$

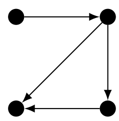
In practice we more often think of a drawing like this:



Technically, this is what is called a *labelled graph*, but we often omit the labels. When we say something about an unlabelled graph like , we mean that the statement holds for any labelling of the vertices.

Here are two examples of related objects that in this course we do not consider graphs:





The first is a *multigraph*, which can have multiple edges and loops; the corresponding definition would allow the edge set and the edges to be multisets. The second is a *directed graph*, in which every edge has a direction; in the corresponding definition the edges would be ordered pairs instead of two-element subsets. In this course we mostly avoid these variants for simplicity, although they are certainly very useful objects. Most facts about graphs in our sense have analogues for multigraphs or directed graphs, although those are often a bit less nice. Another type of graph that we are avoiding is *infinite graphs*; many facts about finite graphs do not extend to infinite graphs.

Some notation: Given a graph G, we write $V(G)$ for the vertex set, and $E(G)$ for the edge set. For an edge $\{x, y\} \in E(G)$, we usually write xy, and we consider yx to be the same edge. If $xy \in E(G)$, then we say that $x, y \in V(G)$ are *adjacent* or *connected* or that they are *neighbors*. If $x \in e$, then we say that $x \in V(G)$ and $e \in E(G)$ are *incident*.

Definition (Subgraphs)**.** *Two graphs* G, G' *are* isomorphic *if there is a bijection* $\varphi : V(G) \to V(G')$ *such that* $xy \in E(G)$ *if and only if* $\varphi(x)\varphi(y) \in E(G')$. *A graph* H *is a* subgraph *of a graph* G, *denoted* $H \subset G$, *if there is a graph* H' *isomorphic to* H *such that* $V(H') \subset V(G)$ *and* $E(H') \subset E(H)$.

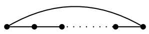
With this definition we can for instance say that is a subgraph of . As mentioned above, when we talk about graphs we often omit the labels of the vertices. A more formal way of doing this is to define an *unlabelled graph* to be an isomorphism class of labelled graphs. We will be somewhat informal about this distinction, since it rarely leads to confusion.

Definition (Degree)**.** *Fix a graph $G = (V, E)$. For $v \in V$, we write*

$$N(v) = \{w \in V : vw \in E\}$$

for the set of neighbors of v (which does not include v). Then $d(v) = |N(v)|$ is the degree *of v. We write $\delta(G)$ for the minimum degree of a vertex in G, and $\Delta(G)$ for the maximum degree.*

Definition (Examples)**.** *The following are some of the most common types of graphs.*

- Paths *are the graphs P_n of the form* *. The graph P_n has n edges and $n+1$ vertices; we call n the* length *of the path.*
- Cycles *are the graphs C_n of the form* *. The graph C_n has n edges and n vertices; we call n the* length *of the cycle.*
- *The* complete graphs *are the graphs K_n on n vertices in which all vertices are adjacent. The graph K_n has $\binom{n}{2}$ edges. For instance, K_4 is* .
- *The* complete bipartite graphs *are the graphs $K_{s,t}$ with a partition $V(K_{s,t}) = X \cup Y$ with $|X| = s, |Y| = t$, such that every vertex of X is adjacent to every vertex of Y, and there are no edges inside X or Y. Then K_{st} has st edges. For example, $K_{2,3}$ is* .

The following are the most common properties of graphs that we will consider.

Definition (Regular)**.** *A graph G is k-*regular *if $d(v) = k$ for all $v \in V(G)$.*

Definition (Bipartite)**.** *A graph G is* bipartite *if there is a partition $V(G) = X \cup Y$ such that every edge of G has one vertex in X and one in Y; we call such a partition a* bipartition.

Definition (Connected)**.** *A graph G is* connected *if for all $x, y \in V(G)$ there is a path in G from x to y (more formally, there is a path P_k which is a subgraph of G and whose endpoints are x and y).*
A connected component *of G is a maximal connected subgraph of G (i.e., a connected subgraph that is not contained in any larger connected subgraph). The connected components of G form a partition of $V(G)$.*

2 Basic facts

In this section we prove some basic facts about graphs. It is a somewhat arbitrary collection of statements, but we introduce them here to get used to the terminology and to see some typical proof techniques.

Proposition 2.1. *In any graph G we have* $\displaystyle\sum_{v \in V(G)} d(v) = 2|E(G)|$.

Proof. We double count the number of pairs $(v, e) \in V(G) \times E(G)$ such that $v \in e$. On the one hand, a vertex v is involved in $d(v)$ such pairs, so the total number of such pairs is $\sum_{v \in V(G)} d(v)$. On the other hand, every edge is involved in two such pairs, so the number of pairs must equal $2|E(G)|$. □

This fact is sometimes called the "handshake lemma" because it says that at a party the number of shaken hands is twice the number of handshakes. It has useful corollaries, such as the fact that the number of odd-degree vertices in a graph must be even.

The next lemma gives a condition under which a graph must contain a long path or cycle. Note that "contains a path" means that the graph has a subgraph that is isomorphic to some P_n, and similarly for cycles. The proof is an example of an *extremal argument*, where we take an object that is extremal with respect to some property, and show that this extremality implies some other property of the object.

Proposition 2.2. *A graph G with minimum degree $\delta(G) \geq 2$ contains a path of length at least $\delta(G)$ and a cycle of length at least $\delta(G) + 1$.*

Proof. Let $v_1 \cdots v_k$ be a maximal path in G, i.e., a path that cannot be extended. Then any neighbor of v_1 must be on the path, since otherwise we could extend it. Since v_1 has at least $\delta(G)$ neighbors, the set $\{v_2, \ldots, v_k\}$ must contain at least $\delta(G)$ elements. Hence $k \geq \delta(G)+1$, so the path has length at least $\delta(G)$.

To find a long cycle, we continue the proof above. The neighbor of v_1 that is furthest along the path must be v_i with $i \geq \delta(G) + 1$. Then $v_1 \cdots v_i v_1$ is a cycle of length at least $\delta(G) + 1$. □

Note that in general these bounds cannot be improved, because $K_{\delta+1}$ has minimum degree δ, but its longest path has length δ and its longest cycle has length $\delta + 1$. In Problem Set 1 we will see that we can find longer paths in graphs that are not complete.

The following lemma can be helpful when trying to prove certain statements for general graphs that are easier to prove for bipartite graphs. The lemma says that you don't have to remove more than half the edges of a graph to make it bipartite. The proof is an example of an *algorithmic proof*, where we prove the existence of an object by giving an algorithm that constructs such an object.

Proposition 2.3. *Any graph G contains a bipartite subgraph H with $|E(H)| \geq |E(G)|/2$.*

Proof. We prove the stronger claim that G has a bipartite subgraph H with $V(H) = V(G)$ and $d_H(v) \geq d_G(v)/2$ for all $v \in V(G)$. Starting with an arbitrary partition $V(G) = X \cup Y$ (which need not be a bipartition for G), we apply the following procedure. We refer to X and Y as "parts". For any $v \in V(G)$, we see if it has more edges to X or to Y; if it has more edges that connect it to the part it is in than it has edges to the other part, then we move it to the other part. We repeat this until there are no more vertices v that should be moved.

There are at most $|V(G)|$ consecutive steps in which no vertex is moved, since if none of the vertices can be moved, then we are done. When we move a vertex from one part to the other, we increase the number of edges between X and Y (note that a vertex may move back and forth between X and Y, but still the total number of edges between X and Y increases in every step). It follows that this procedure terminates, since there are only finitely many edges in the graph. When it has terminated, every vertex in X has at least half its edges going to Y, and similarly every vertex in Y has at least half its edges going to X. Thus the graph H with $V(H) = V(G)$ and $E(H) = \{xy \in E(G) : x \in X, y \in Y\}$ has the claimed property that $d_H(v) \geq d_G(v)/2$ for all $v \in V(G)$. □

The last lemma gives a characterization of bipartite graphs. An "odd cycle" is just a cycle whose length is odd. Again we give an algorithmic proof.

Proposition 2.4. *A graph is bipartite if and only if it contains no odd cycle.*

Proof. Suppose that G is bipartite with bipartition $V(G) = X \cup Y$, and that $v_1 \cdots v_k v_1$ is a cycle in G, with $v_1 \in X$. We must have $v_i \in X$ for all odd i and $v_i \in Y$ for all even i. Since v_k is adjacent to v_1, it must be in Y, so k must be even and the cycle is not odd.

Suppose we have a connected graph G which has no odd cycles. We can obtain a bipartition using the following algorithm. Start with X, Y being empty sets. Pick an arbitrary vertex x_0 and put it in X. Put all the neighbors of x_0 in Y. For each $y \in Y$, put into X all neighbors of y that have not yet been assigned. Then for each $x \in X$, put into Y all neighbors that have not yet been assigned. Keep repeating this until all vertices have been assigned. This algorithm is well-defined because no vertex is assigned more than once (thanks to the stipulation that we only consider unassigned vertices). It remains to be shown that the algorithm terminates (i.e., it does not go on endlessly), and that the resulting partition is really a bipartition of $V(G)$.

The algorithm terminates because G is connected (by assumption). Indeed, this means that every $y \in Y$ has a path $yv_1 \cdots v_k x_0$ to x_0, and in every step at least one more vertex from this past must get assigned.

Next we show that $V(G) = X \cup Y$ is a bipartition. Suppose that two vertices $x_1, x_2 \in X$ are adjacent. By construction, there is a path P from x_1 to x_0 that uses only edges between X and Y, and similarly there is such a path P' from x_2 to x_0; note that these paths may intersect, so their union might not be a path. Let x_3 be the first vertex where P intersects P' (this could be x_0). Then we get a path P'' from x_1 to x_3 to x_2 that also has the property that all its edges are between X and Y. Since this path goes from X to X using only edges between X and Y, it must have an even number of edges. Thus, if we combine it with x_1x_2, we get an odd cycle, which is a contradiction. Similarly, if $y_1, y_2 \in Y$ are adjacent, combining this edge with the path from y_1 to y_2 gives an odd cycle. This shows that all edges of G are between X and Y, so G is bipartite.

We did the above under the assumption that G is connected. If it is not, we can apply the above to each connected component, and arbitrarily combine the bipartitions of the components to get a bipartition of G. $\square$