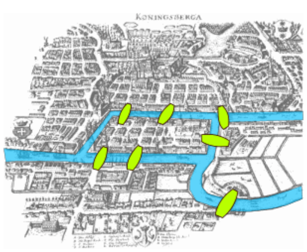
Discrete Mathematics 2015: Lecture I

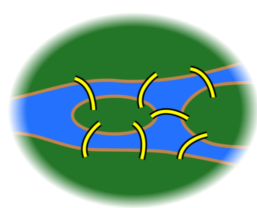
Contents

1 A little bit of history

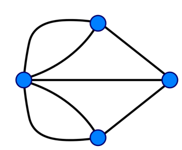
The beginning of graph theory starts with with Euler (1707–1783). The citizens of the city of Königsberg asked him to resolve their favorite pastime question: is it possible to traverse all the 7 bridges joining two islands in the River Pregel and the mainland, in such a way that each bridge is crossed once and only once? Euler represented the problem as a graph (in fact a multigraph), where each vertex corresponded to land regions and edges correspond to bridges connecting these regions. We call this a multigraph because more than one edge can join the same pair of vertices. Solving the Knigsberg Bridge Problem is the same as determining whether it is possible to find a walk in the graph that uses each edge exactly once see Fig. 1 [*]



(a) Konigsberg's bridges.



(b) Konigsberg's bridges.



(c) The corresponding graph representation.

Figure 1: The 7 bridges of Konigsberg Problem.

A similar problem was considered by Sir William Rowan Hamilton (1805-1865) whom even developed a toy based on finding a path visiting all cities in a graph exactly once and sold it to a toy maker in Dublin. Although graphs were implicitly used for solving many games, problems and puzzles, the area of graph theory

[*]Figures taken from Wikipedia.

started when the Danish mathematician Julius Petersen (1839-1910) wrote a purely theoretical article that introduced the mathematical objects that he called *graphs*. Nowadays graphs are used in a lot of contests: e.g. in search engines for ranking web pages, in analysing networks (social, biological, etc), GPS, designing airplane routes etc. In the next section we will see a formal definition as some basic terminology in graph theory.

2 Introduction to graphs

A graph $G = (V, E)$ consists of a finite set of vertices V and a finite set of edges E. An edge represents a binary relation between the vertices. Although there are many varieties of graph concepts studied in the literature, two main ones will be used throughout this course. These correspond to graphs whose edges are directed or undirected. Graphs with directed edges are called directed graphs or simply, digraphs. Graphs with undirected edges are called undirected graphs. In an *undirected graph* an edge *unordered* pair of vertices $\{u, v\}$. While in a *directed graph* an edge is an *ordered* pair of vertices (u, v). Observe that you can have directed edges of the form (a, a), these edges are called *loops*. In this course we will consider only directed graphs without loops, hence we will consider only edges (u, v) with distinct u, v.
We will say that an edge (u, v) is *incident* to u and v and the vertices u and v are called *adjacent*.
A graph is usually depicted visually, by drawing the elements of the vertices set as boxes or circles, and drawing the elements of the edge set as lines or arcs between the boxes or circles. In Fig. 2 we present two different drawings of the same graph $G = (V, E)$ with $V = \{a, b, c, d, e\}$ and $E = \Big\{\{a, b\}, \{b, c\}, \{c, d\}, \{d, e\}, \{a, e\}\Big\}$.

Figure 2: The same graph represented in two different drawings.

A graph that can be drawn in a plan such that the edges intersect only in the endpoints is called *planar*. The graph in Fig. 2 is planar as its representation on the left has no crossing edges.
Given a graph G, the *degree* of a vertex v in G (denoted by $d_G(v)$) is the number of edges incident to it. In symbols for a vertex $v \in V$ the degree $d_G(v)$ is defined as $d_G(v) = |\{u : \{u, v\} \in E\}|$.
If G is directed we define the *out-degree* (*in-degree*) of of a vertex v as follows:

$$d^+_G(v) = |\{u : (v, u) \in E(G)\}| \qquad \text{and} \qquad d^-_G(v) = |\{u : (u, v) \in E(G)\}|$$

For a graph $G = (V, E)$ we usually refer by n to the number of vertices and m the number of edges. In other words, $n = |V|$ and $m = |E|$.

Exercise 2.1 *For a graph $G = (V, E)$ on n vertices (i.e. $|V| = n$) what is the minimum and the maximum degree that a vertex can have?* Ans.

A *walk* in a graph is a sequence $v_1, e_1, \ldots, e_{k-1}, v_k$ of alternating vertices and edges, beginning and ending with vertices, such that for each $1 \leq i \leq k$ the edge e_i has endpoints v_i and v_{i+1}.
A *path* is a walk with no repeated vertices.
A *cycle* is a path starting and ending in the same vertex. Observe that the graph depicted in Fig. 2 is a cycle.
For a directed graph we can define a directed walk, path and directed cycle in a similar way with the only difference that we have to respect edge direction.

Connectivity Imagine a map of a country in which cities are connected by roads, railways, etc. Its main purpose is to show whether and how to travel from one place to another. This is related to an important graph property: the connectivity.
Two vertices u and v in a graph G are said *connected* if there exists a path starting from u and ending in v in G. A graph is connected if every two of its vertices are connected.

Exercise 2.2 *Show that deleting any edge that belongs to a cycle in a graph does not disconnect the graph.*

There are two questions that arise naturally: (1) How can we define the connectivity in directed graphs, and (2) Is it possible to consider connectivity in such a way one graph is more connected than the other?

A graph that is not connected is called *disconnected*. A disconnected graph can be “broken” into connected graphs called *components*.

Given a directed graph D we say it is *weakly connected* if the underlying graph (that is the graph obtained from D by ignoring the direction of the edges) is connected. We say that the graph D is *strongly connected* if for every two vertices u and v of D, there is a directed path from u to v and a directed path from v to u.

Exercise 2.3 *Can you think of a real life example where the strong connectivity of a graph is required?*

Consider the graphs in Fig. 3. Intuitively one may see that they have not the same “degree of connectivity”. One way to formalize this is to say that a graph is “more connected” than another if you have to remove more edges (or vertices) in order to disconnect it. More formally we will say that a graph G is k-edge connected (k-vertex connected) if we need to remove at least k edges (vertices) in order to disconnect the graph. An edge e whose removal disconnects the graph is called a *bridge*. In Fig. 3.c the edge $\{b, e\}$ is a bridge. The vertex whose removal disconnects the graph is called an *articulation* point. In Fig. 3.c the vertex b, e are articulation points.

Subgraphs of a graph Consider $G = (V, E)$ we say that $G' = (V', E')$ is a *subgraph* of G if $V' \subseteq V$ and $E' \subseteq E$. *A spanning subgraph* of G is a subgraph that is connected and contains all the vertices of G. The graph $G' = (V', E')$ is an *induced subgraph* of G if $V' \subseteq V$ and E' contains all the edges in E whose endpoints are in V'. For an example see Fig. 4.

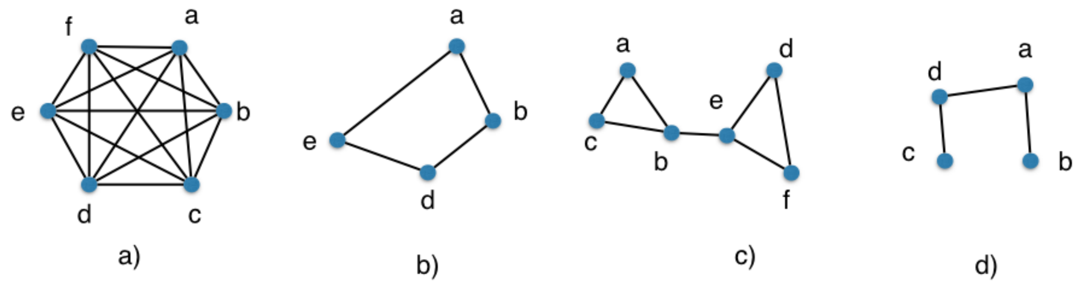


Figure 3: a) A $n-1$-edge connected graph, b) A 2-edge connected graph, c) A 1-edge connected graph, d) A 1-edge connected graph

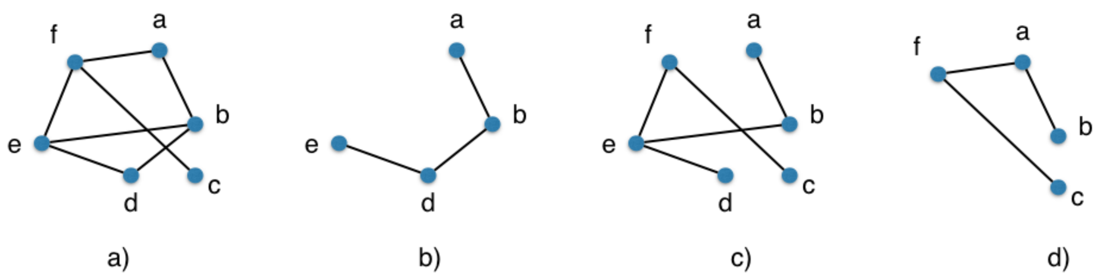


Figure 4: a) A graph G, b) A subgraph of G, c) A spanning subgraph of G, d) An induced subgraph of G.

Important graphs Here we see some important types of graphs. An *empty* graph is a graph with no vertices and no edges (that is $E = V = \emptyset$). A *complete* graph (sometime also called a *clique*) is a graph that has all the edges between vertices. A complete graph on n vertices is usually denoted by K_n. A graph that has no cycle is called *acyclic*.

Exercise 2.4 *How many edges does K_n have? Ans.*

Imagine a city in winter where all the streets were blocked by snow. To assure the travelling the streets have to be cleaned. However, the city cannot afford economically to clean all of them, however it wants to clean all the streets so that it is possible to travel by vehicle between all places in the city along cleaned streets. How can we do this?

Clearly this problem can be modeled as a graph. We want to find a subgraph that is connected and has no cycle. This because as we have seen in Exercise 2.2 deleting an edge from a cycle does not disconnect the graph. Hence, the street that corresponds to that edge is not necessary useful for reaching some place. In fact to find the best solution we have to look for a spanning subgraph that is acyclic and connected. This is called a tree.
A graph that is connected and has no cycle is called a *tree*. We will usually denote a tree by T rather than by G. Sometimes we will need to distinguish a vertex in a tree, in that case we will talk about *rooted trees* and the distinguished vertex will be called *a root*. The *level* of a vertex in a rooted tree is the number of edges from that vertex to the root. The *height* of a rooted tree is the maximum level of any vertex in the tree.
A *leaf* is a vertex of degree 1.

Exercise 2.5 *Show that every treee has a leaf. Ans.*

Another interesting graph is the bipartite graph.
A *bipartite* graph is a graph $G = (V_1, V_2, E)$ whose set of vertices can be partitioned into two sets V_1 and V_2 ($V_1 \cap V_2 = \emptyset$) and for every $e = \{u, v\} \in E$ we have $v \in V_1$ and $u \in V_2$. In other words a graph is bipartite if it is possible to partition its vertex sets into two sets such that all the edges are across the sets.

3 Basic results on graphs

In proving results in discrete mathematics several useful combinatorial rules or combinatorial principles are commonly used.

- *Proof by contradiction:* The basic idea is to assume that the statement we want to prove is false, and then show that this assumption leads to an absurd. We can then conclude that we were wrong to assume the statement was false, so the statement must be true.

- *The pigeonhole principle:* The basic idea is that if n pigeons (or items) are put into m pigeonholes (or containers), with $n > m$, then at least one hole (container) must contain more than one pigeon (item).

- *The Double Counting Principle:* If the same set is counted in two different ways, you get the same answer.

Have you heard about the "Domino Effect" (see Fig. 5)? Step 1. The first domino falls. Step 2. When a domino falls, the next domino falls. Then we conclude that ... all dominos will fall! This perfectly illustrates the third technique of proof, that is the mathematical induction.

- *Induction:* We use this type of proof if we need to proof that the statements $S_1, S_2, S_3, S_4, \ldots$ are all true. This technique consists of two steps:
  - Prove that the first statement S_1 is true.
  - Given any integer $k \geq 1$, prove that if the statement S_k is true then the statement S_{k+1} is also true. It follows by mathematical induction that every S_n is true.

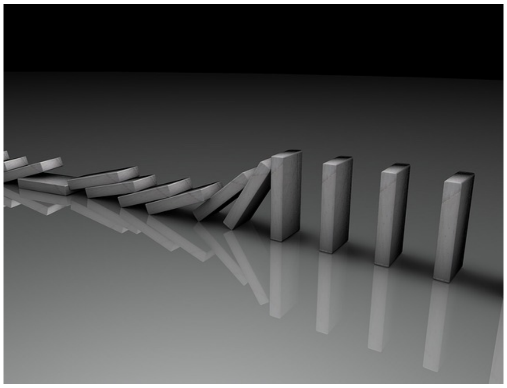


Figure 5: The domino effect.

We will see now how to apply these techniques in order to prove some first results in graph theory.

Claim 3.1 *Every connected graph contains two vertices of the same degree.*

Proof: Observe that in any connected graph on n vertices, the minimum degree of a vertex is 1 and the maximum is $n-1$. That means that for any vertex there are $n-1$ possible values for its degree. As there are n vertices then by the The pigeonhole principle two vertices must have the same degree. □

Theorem 3.1 *For a graph $G = (V, E)$ with $V = \{v_1, \ldots, v_n\}$ it holds*

$$\sum_{i=1}^{n} d_G(v_i) = 2|E|$$

Proof: To prove this theorem we will use the technique of *Double-counting*. Observe that each edge contributes exactly twice in the sum. Based on this observation we consider the following set:

$$\mathcal{M} = \{(v, e) : v \in V, e \in E \text{ and } v \text{ is incident to } e\}$$

Now, we will count the elements of $\mathcal{M}$ in two different ways. First, we count the pairs (v, e) by considering first v. Note that M is then defined as the following disjoint union:

$$M = \cup_{v \in V}\{(x, e) : e \in E \text{ and } v \text{ is incident to } e\}$$

implying that

$$|M| = \sum_{i=1}^{n} d_G(v_i)$$

Second, we fix the second element e and we have

$$M = \cup_{e \in E}\{(x, e) : v \in V \text{ and } v \text{ is incident to } e\}$$

implying that

$$|M| = \sum_{e \in E} 2 = 2|E|$$

This concludes the proof. □

Exercise 3.1 *For a graph $G = (V, E)$ show that the number of vertices of odd degree is even.* Ans.

Try now to show the following result.

Theorem 3.2 *A graph G is a tree if and only if every two vertices of G are connected by only one path.*

Theorem 3.3 *A tree T on n vertices has exactly $n-1$ edges.*

Proof: We will show this proof by induction on the number of vertices of the tree. Base case: $n = 1$ is trivially true as a tree T on 1 vertex has 0 edges. Suppose now that every tree that has k vertices has exactly $k-1$ edges. Consider now a tree T on $k+1$ vertices. By Exercise 3.1 there is a vertex v in T which is a leaf. Let T' be the graph obtained from T removing the leaf v. Note that $|E(T')| = |E(T)| - 1$. As v was a leaf, T' is connected and has no cycle (the removal of a vertex does not create cycles). Hence, T' is a tree on k vertices and by the induction hypothesis it has $k-1$ edges. Finally, T has one edge more than T', i.e. k edges. This concludes the proof. □

Is it true that if $G = (V, E)$ is a graph on n vertices that has $n-1$ edges then G is connected?

Exercise 3.2 *What is the maximum number of edges that a disconnected graph can have?* Ans.

4 Representation of Graphs: Adjacency Matrix and Adjacency List

The two main graph representations we use when talking about graph problems are the adjacency list and the adjacency matrix.

Adjacency Matrix A graph $G = (V, E)$ on n vertices can be represented as a $n \times n$ matrix M such that $M[i, j] = 1$ if $\{v_i, v_j\} \in E$ and $M[i, j] = 0$ otherwise.
Observe that an adjacency matrix takes up $\Theta(n^2)$ storage.

Adjacency List An adjacency list is a vector of lists. Each cell in the vector corresponds to a vertex v and contains a list of edges $\{u, v\}$ that are incident to u. Thus, an adjacency list takes up $\Theta(n + m)$ space.

Exercise 4.1 *Try to answer the followings:*

- *Thinking in terms of storage when is more convenient to store a graph as an adjacency list? an adjacency matrix?*
- *How much time does it take to check whether an edge is present in a graph, if the graph is stored using: (a) adjacency list, (b) adjacency matrix?*

5 Solution of the exercises

Solution 1 *Exercise 2.1* 0 *and* $n - 1$.

Solution 2 *Exercise 2.4* $n(n - 1)/2$.

Solution 3 *Exercise 2.5*

Solution 4 *Exercise 3.1 It follows as the sum of all degrees must be even.*

References

[1] J.A. Bondy and U.S.R. Murty, Graph Theory with Applications, North-Holland, 1976. (This book is out of print. You can download their personal copy from the web page: http://book.huihoo.com/pdf/graph-theory-With-applications/pdf/GTWA.pdf)

[2] R. Diestel, Graph Theory, 4th edition, Springer, 2010. See electronic edition at: http://diestel-graph-theory.com/