Foreword

Graphs are an important and burgeoning topic in Combinatorics, not only in research, but also in mathematical olympiads. They have a strong presence in the International Mathematical Olympiad, National Team Selection Tests and sometimes National Olympiads as well.

Yet, when asked by an aspiring problem solver where to learn graphs from, we inevitably hesitate to respond. There are very good books on graphs, such as Diestel's excellent *Graph Theory*, but, being written as advanced textbooks, they are not really suitable for our problem solver. They quickly dive into complex topics, assume a fair amount of pre-requisite knowledge, and do not contain many problems. There are also some good combinatorics books for olympiads, but they do not to present the basics of graphs in the structured way in which they should be presented.

So we thought of writing a book to bridge this gap between the enthusiastic problem solver and the beautiful field of graphs. We are going carefully through the basics, but without throwing too much theory at the reader. The guiding principle for writing this book was not to show more than needed: for a method or idea, we tried to present a reasonably easy example, leaving the harder ones for the reader to solve.

We also aimed at a very friendly and informal book, with a lot of comments, drawings, and most of all, intuitions from someone who has been, not too long ago, in the position of the reader. We often gave more than one solution to a problem, so it is worth checking after having solved a problem. The source and author of a problem, if known, are also mentioned at the end.

The book has an organic structure, with chapters in their natural order. We kept the theory part of the chapters within reasonable limits in order to encourage the reader to read through carefully before proceeding to the problems. Our recommendation would then be to try to go through the book in order, reading the theory with patience. It is also worth trying to prove a theorem by oneself before reading the proof. Alternatively, if the reader wants, she could also read each chapter, work on most of the problems, and then go to the next chapter, leaving the hardest problems for after she finished all chapters.

In graph theory, unlike, say, in geometry, there isn't a clear distinction between theorems and problems. There is hardly a theorem that can be applied over and over again, while many of the problems could be regarded as mini-theorems. For this reason, many of the results proven in this book are noncommittally labelled 'Proposition', a term which captures precisely this ambiguity. Anyway, we would like to encourage the reader to join into this way of thinking, and not to look for theorems and applications of theorems.

We tried to assume as little as possible, the only things being made use of in the ten chapters being some basic principles in combinatorics, such as the pigeonhole principle, and some elementary inequalities, such as AM-GM. We did add two appendices though, one on the probabilistic method in graphs, and the other on linear algebra in graph theory. Both assume a good knowledge of probabilities and linear algebra, so they should not be used to actually learn about these topics. Also, we encourage the reader to resist the temptation, common nowadays, to try to learn more without having mastered the basics. The appendices are for those who have already mastered the ten chapters.

There is a common practice in mathematical competitions to avoid graphs terminology, in particular the very word 'graph'. To achieve this, graphs problems, that presumably came to their authors as graphs problems, are often painted over with a story involving airlines, roads, and what not. A reason for this is that it is not a pre-requisite for olympiads to know graphs. To us, this sounds a bit flimsy: it is unlikely that someone who has never heard of graphs would solve a hard graphs problem, even if it is formulated in terms of airlines. So, in this book, we have reformulated the problems in graph theoretical terms, to make everything neater and easier to read. The only exceptions are those in which the interpretations actually motivate the problem and make it more intelligible.

We wrote this book, our first one, as we would have liked to read when in high school. Hopefully, things haven't changed too much.

Contents

Chapter 1

Introduction

Graphs can be seen as representing relations in the world. Suppose we have a group of people, some pairs of them knowing each other, some not. If 'knowing each other' is, as the English language seems to have it, a mutual relation, we can try to represent these relations by some drawing points on the paper to represent the people, and connecting pairs that know each other. It is irrelevant whether we connect two points with a straight line or not. In this way, we might be able to visualise better how many friends each person has, or perhaps how many common friends two certain people have.

Or suppose that we have a number of cities, and two-way airline services between them. Again, we might try to represent the pairs that have a service between them, perhaps to see how to get from one city to another with as few changes as possible. We get the same kind of drawing.

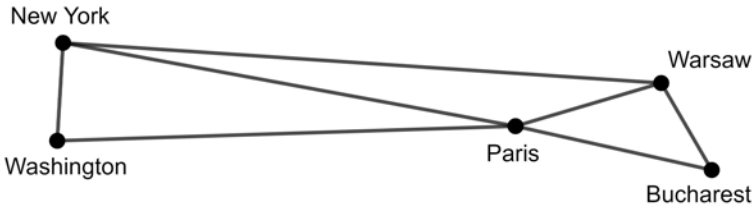


Figure 1.1: The airline services between New York, Washington, Paris, Warsaw and Bucharest

These, and many other, types of relations can be represented by a graph. Yet once we define what a graph is, and start to ask questions about graphs, they will hopefully prove so fascinating in their own right as to make the reader forget all the real-world interpretations from which we started.

Definition 1.1. An (undirected) *graph* G consists of a set V, called the vertex set, and a set E of edges between the vertices (formally, pairs $\{u, v\}$, where u and v are elements of V). We write $G(V, E)$.

Graphs can be finite or infinite, depending on whether V is finite or infinite. Unless otherwise stated, we assume graphs to be finite.

A graph is *simple* if between every two vertices there is at most one edge and there are no *loops* (edges between v and itself). The formal definition we gave implies that graphs are simple, but more complex definitions could render them otherwise.

Note that, even if we often represent the graph on paper, as described before, the graph is not a geometrical object. There are many ways in which we can 'draw' a graph.

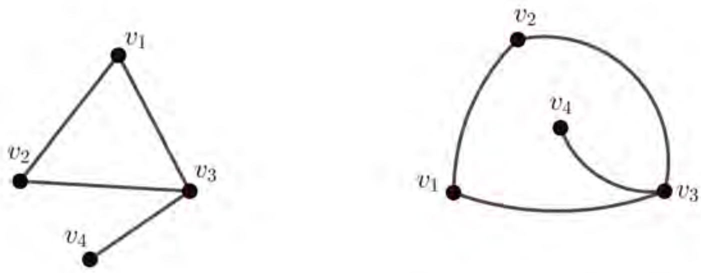


Figure 1.2: Two drawings of the same graph

There are many definitions in graph theory, many of them natural. We will give most of them now, to be then able to speak graphese at will.

Definition 1.2. In a graph $G(V, E)$:

- For an edge uv, u and v are called its *endpoints*

  Two vertices are called *adjacent* if there is an edge between them. We also say that they are *connected* or that they are *neighbours*.

  Two edges are called *adjacent* if they share a common endpoint.

  An edge is called *incident* on a vertex if it has that vertex as an endpoint.

- The *degree* of a vertex $v \in V$ is the number of edges incident on v (i.e. having v as an endpoint). It is typically denoted by $d(v)$.

- The maximal degree of a graph G is typically denoted by $\Delta(G)$.

- The minimal degree of a graph G is typically denoted by $\delta(G)$.

We will now define some important objects that have a more global importance in the graph. Suppose one wants to plan a trip, going from city to city by airplane. This is what we will call a trail. Additional conditions, such as not going through the same city twice, give rise to new notions.

Definition 1.3. In a graph G:

- A *trail* from v to w is a sequence of vertices $v = v_1, v_2, \ldots, v_n = w$ such that $v_i v_{i+1} \in E$ for $i = 1, 2, \ldots, n-1$.

  A *walk* from v to w is a trail in which all edges are distinct.

  A *path* from v to w is a trail in which all vertices are distinct.

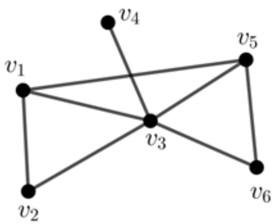
- The *length* of a trail/walk/path is the number of (not necessarily distinct) edges in it.

- The *distance* between two vertices u and v, usually denoted by $d(u, v)$, is the length of the shortest path between u and v.

- A *closed trail* is a sequence of vertices $v_1, v_2, \ldots, v_n$ such that $v_i v_{i+1} \in E$ for $i = 1, 2, \ldots, n$, with indices taken cyclically. To emphasise that a closed trail returns to v_1 we will often denote a closed trail by $v_1, v_2, \ldots, v_n, v_1$.

  A *circuit* is a closed trail in which the edges $v_i v_{i+1}$ are pairwise distinct.

  A *cycle* is a circuit in which the vertices are pairwise distinct.



$v_1, v_3, v_5, v_6, v_3, v_4$ is a walk, but not a path.
v_1, v_2, v_3 is a path.
$v_1, v_2, v_3, v_5, v_6, v_3, v_1$ is a circuit, but not a cycle.
v_1, v_2, v_3, v_1 is a cycle.

Figure 1.3: Walk, paths, cycles, circuits

- The *length* of a circuit/cycle is the number of edges in it.

  A cycle of length 3 is often called a *triangle*.

- The *girth* of a graph is the length of the minimal cycle (if one exists, otherwise it is ∞).

Definition 1.4. A graph is *connected* if for any two vertices v and w, there is a path from v to w.

Remark. (Connected components) We observe that graphs that are not connected are essentially 'a collection of connected graphs'.

We will call the maximal connected subgraphs of a graph G the *connected components* of G.

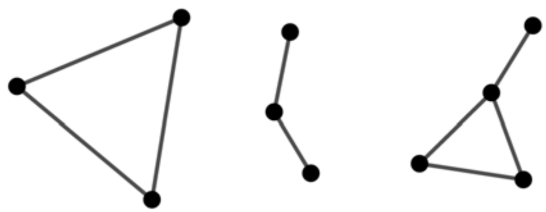


Figure 1.4: A graph with 3 connected components

There are some special kinds of graphs that we will use:

Definition 1.5. We have the following types of graphs:

- A *tree* is a connected graph with no circuits.

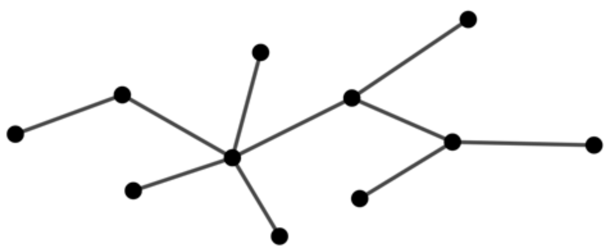


Figure 1.5: A tree

- Given a graph G, its complement $\overline{G}$ consists of the same vertices and exactly the edges that are not edges in G.

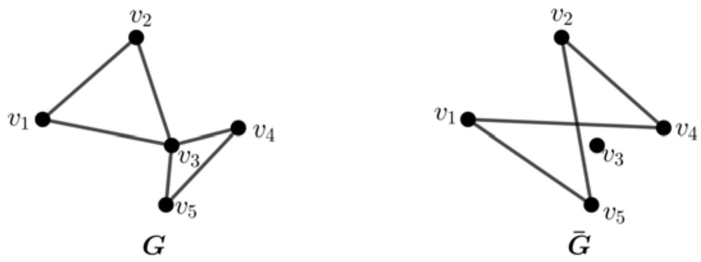


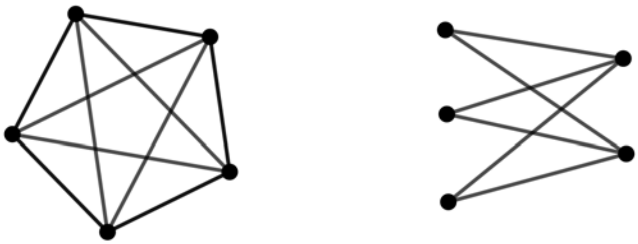
Figure 1.6: A graph G and its complement $\overline{G}$

- A graph is called *regular* if all vertices have the same degree (k-regular if all degrees are k).

- K_n is the *complete graph* on n vertices, i.e. the graph with all possible edges between n vertices.

- A graph is called *bipartite* if its vertices can be partitioned into sets A and B such that all edges are between a vertex in A and one in B.

- $K_{m,n}$ is the *complete bipartite graph*, i.e. the bipartite graph with sets A and B such that $|A| = m, |B| = n$ and all the possible edges between A and B are drawn.



$\boldsymbol{K_5}$ $\boldsymbol{K_{3,2}}$

Figure 1.7: K_5 and $K_{3,2}$

And now some definitions about the relations between graphs:

Definition 1.6. We have the following definitions:

- Two graphs are called *isomorphic* if 'they are the same graph'.

  (Formally, they are isomorphic if there is a bijection f between the vertices of the two graphs such that uv is an edge if and only if $f(u)f(v)$ is an edge.)

- A *subgraph* of a graph G is a subset of the vertices of G together with some the edges between them that were in G.

  A subgraph is called *induced* if it contains all the edges between its vertices that were edges in G.

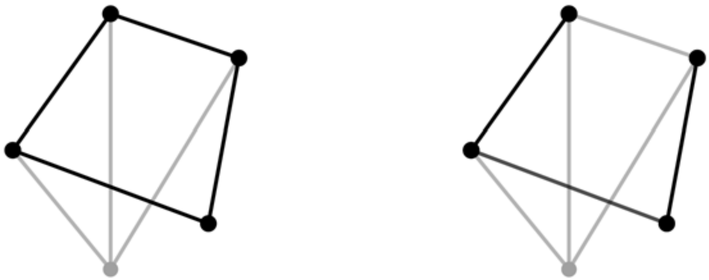


Figure 1.8: An induced and a non-induced subgraph on 4 vertices

A subgraph is called *spanning* if its set of vertices is the set of vertices of G.

As an introduction to graphs, we shall ask and answer some basic questions about graphs, yet chosen in such a way as to illustrate the basic methods of graph theory. These are the building blocks that we will use in later chapters.

The problems in this introduction are meant not so much to develop theory (this we will do later), as to get us used to graphs and to train our intuitions.

Pigeonhole Principle

The pigeonhole principle is the basic observation that if we have $nk + 1$ objects in n sets, then there is a set with at least $k + 1$ objects.

A classical problem in graphs is that if we have six people, either three of them are pairwise friends, or three of them are pairwise strangers. Rephrased in graph theoretical terms, we have the following:

Proposition 1.7. *In any graph on* 6 *vertices, there exist three vertices that are pairwise connected, or three vertices that are pairwise not connected.*

Proof. Pick a vertex, say v. By the pigeonhole principle, there are three other vertices that are either all connected or all not connected to v. Assume the former.

If two of these are connected between themselves, they form a triangle with v. Otherwise, the three of them are pairwise not connected.

Similarly for the case in which the three vertices are not connected to v. □

In general, the pigeonhole principle can come in handy when we have a lot of edges and wanting to find something like a complete subgraph.

A more general version of this problem, and other similar problems, are to be found in the Ramsey theory chapter.

Double Counting

The technique of double counting consists of expressing something in two different ways and then deducing that the results are equal.

Simple as it might sound, the technique of double counting comes in very handy in many areas of combinatorics, including graph theory. This is perhaps to be expected, given the fact that we are dealing with two objects, vertices and edges. The most basic observation is:

Theorem 1.8. *(Handshaking lemma) In every graph $G = (V, E)$,*

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

Proof. The left-hand side counts edges: $d(v)$ counts those edges that are incident on v. But we can easily observe that each edge uv is counted twice, once for $d(u)$, and once for $d(v)$. The conclusion follows.

(Formally, both sides count the number of pairs (v, e), where v is a vertex, and e an edge incident on v. For each v, there are $d(v)$ pairs, while for each e, there are exactly two pairs). □