Graph Theory I

Math 7703

Shagnik Das
National Taiwan University

Fall 2023

Acknowledgements

These course notes have been (very slightly) adapted from Benny Sudakov's notes for his *Graph Theory* course at ETH Zürich, and I am very grateful to him for sharing them with me.

Some of the material in these notes is based on the books *Graph Theory* by Reinhard Diestel, *Introduction to Graph Theory* by Douglas West, and *Proofs from the Book* by Martin Aigner and Günter Ziegler.

Contents

$\underbrace{E_n \cdot (n-1)} \cdot C_2^n$

1 Fundamentals

1.1 Graphs

Definition 1.1. A (simple) *graph* G is a pair $G = (V, E)$ where V is a set of *vertices* and $E \subseteq \binom{V}{2}$ is a set of unordered pairs of vertices. The elements of E are called *edges*. We write $V(G)$ for the set of vertices and $E(G)$ for the set of edges of a graph G. Also, the *order* of a graph $|G| = |V(G)|$ is the number of vertices it contains, while the *size* $e(G) = |E(G)|$ denotes the number of edges.

Remark 1.2. One can also consider *directed graphs*, where the edges $E \subseteq V \times V$ are ordered pairs of vertices, or *multigraphs*, where there may be mutliple edges between a pair of vertices. However, in this course, unless explicitly stated otherwise, we shall only concern ourselves with simple graphs.

Definition 1.3. Some further terminology concerning relations between vertices and edges:

- Vertices u, v are *adjacent* in G if $\{u, v\} \in E(G)$.
- An edge $e \in E(G)$ is *incident* to a vertex $v \in V(G)$ if $v \in e$.
- Edges e, e' are *incident* if $e \cap e' \neq \varnothing$. 相交
- If $\{u, v\} \in E$ then v is a *neighbour* of u.

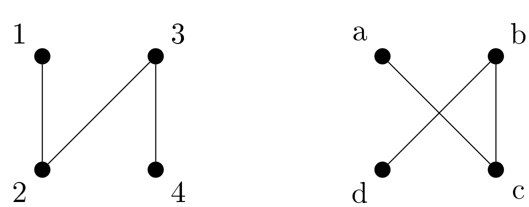
Example 1.4. Any symmetric relation between objects can be modelled with a graph. For example:

- let V be the set of people in a room, and let E be the set of pairs of people who met for the first time today;
- let V be the set of cities in a country, and let the edges in E correspond to roads connecting them;
- the internet: let V be the set of computers, and let the edges in E correspond to the links connecting them.

1.2 Graph isomorphism

Although graphs are formally defined as a set of vertices and a set of edges, simply listing all the edges of a graph is not a very convenient representation for a human to understand. The usual way to picture a graph is to put a dot for each vertex and to join adjacent vertices with lines. Bear in mind, though, that the specific drawing is irrelevant, and all that matters is which pairs are adjacent.

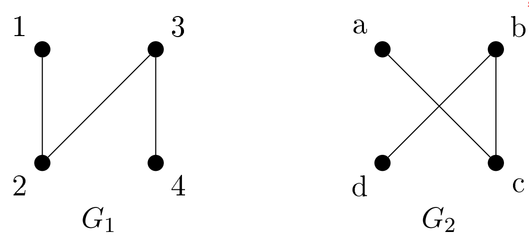
Question 1.5. Consider the two graphs drawn below. How similar can they be said to be?



Definition 1.6. Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be graphs. An *isomorphism* $\phi : V_1 \to V_2$ is a bijection (a one-to-one correspondence) from V_1 to V_2 such that $\{u, v\} \in E_1$ if and only if $\{\phi(u), \phi(v)\} \in E_2$. We say G_1 is *isomorphic* to G_2 if there is an isomorphism between them.

※有邊 → 有邊
點 → 點

Example 1.7. Recall the graphs in Question 1.5:



The function $\phi : G_1 \to G_2$ given by $\phi(1) = a$, $\phi(2) = c$, $\phi(3) = b$, $\phi(4) = d$ is an isomorphism.

Remark 1.8. Isomorphism is an equivalence relation of graphs. This means that

- Any graph is isomorphic to itself 反身性 (reflexive)
- if G_1 is isomorphic to G_2 then G_2 is isomorphic to G_1 (symmetric)
- If G_1 is isomorphic to G_2 and G_2 is isomorphic to G_3, then G_1 is isomorphic to G_3. (transitive)

Definition 1.9. An *unlabelled graph* is an isomorphism class of graphs. In the previous example G_1 and G_2 are different labelled graphs, but since they are isomorphic, they represent the same unlabelled graph.

Since we are only interested in graphs up to isomorphism (that is, we only care about unlabelled graphs), the actual labels of the vertices do not matter, and so we will generally take $V = [n] = \{1, 2, \ldots, n\}$.

1.3 The adjacency and incidence matrices

Although drawings of graphs are a convenient human representation, they suffer from two major drawbacks:

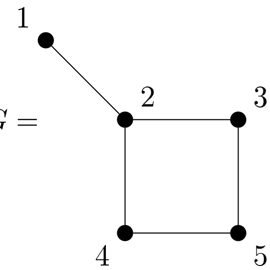
- they can be hard to read when there are many vertices and even more edges, and
- they are not very computer-friendly. XD

We therefore often represent graphs in matrix form, as we describe in this section.

Definition 1.10. Let $G = (V, E)$ be a graph with $V = [n]$. The *adjacency matrix* $A = A(G)$ is the $n \times n$ symmetric matrix defined by

$$a_{ij} = \begin{cases} 1 & \text{if } \{i, j\} \in E, \\ 0 & \text{otherwise.} \end{cases}$$

Example 1.11.

$G =$ 

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}.$$

Remark 1.12. Any adjacency matrix A is real and symmetric, hence the spectral theorem proves that A has an orthogonal basis of eigenvalues with real eigenvectors. This important fact allows us to use *spectral methods* in graph theory. Indeed, there is a large subfield of graph theory called *spectral graph theory*.

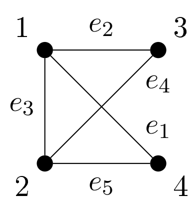
wow!

Aside from the adjacency matrix, we can also represent graphs using the incidence matrix, defined below.

Definition 1.13. Let $G = (V, E)$ be a graph with $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$. Then the *incidence matrix* $B = B(G)$ of G is the $n \times m$ matrix defined by

$$b_{ij} = \begin{cases} 1 & \text{if } v_i \in e_j, \\ 0 & \text{otherwise.} \end{cases}$$

Example 1.14.

$G =$ 

$$B = \overbrace{\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix}}^{E}.$$

Remark 1.15. Every column of B has $|e| = 2$ entries 1.

1.4 Degrees

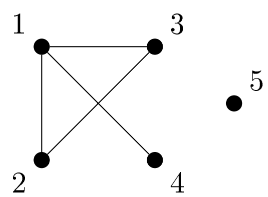
It is often important to take a more local view of a graph, and study what goes on in the immediate surroundings of an individual vertex. The first step in this direction is to look at how many edges a vertex is incident to.

Definition 1.16. Given $G = (V, E)$ and a vertex $v \in V$, we define the *neighbourhood* $N(v) = \{u \in V : \{u, v\} \in E(G)\}$ of v to be the set of neighbours of v. Let the *degree* $d(v)$ of v be $|N(v)|$, the number of neighbours of v. A vertex v is *isolated* if it has no neighbours, that is, if $d(v) = 0$.

Remark 1.17. $d(v)$ is the number of ones in the row corresponding to v in the adjacency matrix $A(G)$ or the incidence matrix $B(G)$.

有連之邊 有連之点

Example 1.18.



$d(1) = 3,\ d(2) = 2,\ d(3) = 2,\ d(4) = 1,\ d(5) = 0;$

5 is isolated.

Question 1.19. For any graph G on the vertex set $[n]$ with adjacency and incidence matrices A and B, show that $\underbrace{BB^T}_{X} = D + A$, where

$$D = \begin{pmatrix} d(1) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & d(n) \end{pmatrix}.$$

neighbours that a vertex has

$A: \begin{pmatrix} 0 & & \\ & 0 & \\ & & \ddots \\ & & & 0 \end{pmatrix}$ is '1' if v_i, v_j are adjacent, '0' if v_i, v_j aren't.

$X_{ij} = \sum_1^n B_{i,\cdot} B^T_{\cdot,j}$

$= \sum_1^n B_{i,\cdot} \times B_{j,\cdot}$ = common edges between v_i and v_j

The degree of a vertex is a local statistic, but looking at the extreme values it takes over all vertices of the graph gives rise to some very important graph parameters.

Notation 1.20. The *minimum degree* of a graph G is denoted

$$\delta(G) = \min\{d(v) : v \in V(G)\},$$

the *maximum degree* is denoted

$$\Delta(G) = \max\{d(v) : v \in V(G)\},$$

and the *average degree* is

$$\bar{d}(G) = \frac{\sum_{v \in G} d(v)}{|V(G)|}.$$

Note that we trivially have $\delta(G) \leq \bar{d}(G) \leq \Delta(G)$ for all graphs G.

Definition 1.21. We say a graph G is *d-regular* if all vertices have degree exactly d.

Although graphs are very flexible structures, it turns out that there are restrictions on what degrees a graph can have. For instance, consider the following question.

Question 1.22. Is there a 3-regular graph on 9 vertices?

To answer the question, we will prove our first result of this course, a simple yet very useful equation between the degrees and the size of a graph.

The sum of degrees must be even.

→ 3×9 is not even.

$e(G) = 27$

$\sum_v d(v) = 27$

$2 = 2 \cdot 1$

of c.

Proposition 1.23 (Handshake Lemma). *For every graph $G = (V, E)$, $\sum_{v \in V} d(v) = 2e(G)$.*

Proof. We double-count the incidences between vertices and edges; that is, we count the set $I = \{(v, e) : v \in V(G), e \in E(G), v \in e\}$. Every edge $e = \{u, v\}$ is counted twice, once in the pair (u, e) and once in the pair (v, e). Thus $|I| = 2|E| = 2e(G)$. On the other hand, a vertex $v \in V(G)$ is counted once for every edge it is incident to, which is precisely its degree $d(v)$. Thus $|I| = \sum_{v \in V(G)} d(v)$, and the result follows. □

Corollary 1.24. *Every graph has an even number of vertices of odd degree.*

This shows that the answer to Question 1.22 is "no".

1.5 Subgraphs

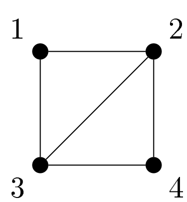
The degree or neighbourhood of a vertex is a very local part of a graph. Another very important concept is that of a subgraph, which is again part of a graph, although potentially involving several, if not all, vertices.

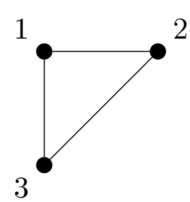
Definition 1.25. A graph $H = (U, F)$ is a *subgraph* of a graph $G = (V, E)$ if $U \subseteq V$ and $F \subseteq \binom{U}{2} \cap E$. If $U = V$ then H is called *spanning*. ←砍邊？保點

Definition 1.26. Given $G = (V, E)$ and $U \subseteq V$ ($U \neq \varnothing$), let $G[U]$ denote the graph with vertex set U and edge set $E(G[U]) = \{e \in E(G) : e \subseteq U\}$. (We include all the edges of G which have both endpoints in U.) Then $G[U]$ is called the subgraph of G *induced* by U.

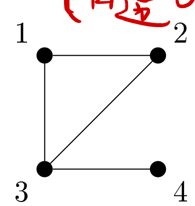
keep adjacency of vertices in U
(有邊的還是有邊)

Example 1.27.

$G =$ 



induced subgraph



not induced
but spanning

1.6 Special graphs

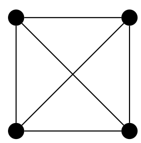
We have so far seen a few arbitrary examples of graphs. Here we collect a few important families of graph that will appear repeatedly in our course.

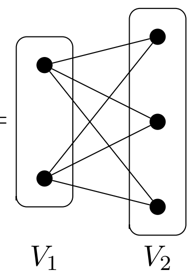
- $K_n = \left([n], \binom{[n]}{2}\right)$ is the *complete graph*, or a *clique*. Take n vertices and all possible edges connecting them.
- An *empty graph* $G = (V, \emptyset)$ has no edges.

point set

- $G = (V, E)$ is *bipartite* if there is a partition $V = V_1 \cup V_2$ into two disjoint sets such that each $e \in E(G)$ intersects both V_1 and V_2 (equivalently, V_1 and V_2 induce empty subgraphs).
- $K_{n,m}$ is the *complete bipartite graph.* Take $n + m$ vertices partitioned into a set A of size n and a set B of size m, and include every possible edge between A and B.

Example 1.28.

$K_4 =$ 

$K_{2,3} =$ 

1.7 Walks, paths and cycles

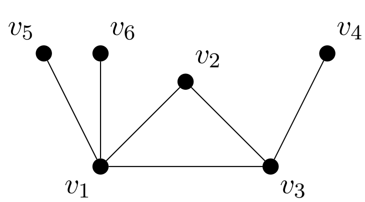
If a graph represents a network, then one crucial consideration is how we can travel from one vertex to another. This is captured through the notion of walks, paths and cycles.

Definition 1.29. A *walk* in G is a sequence of vertices $v_0, v_1, v_2, \ldots, v_k$, and a sequence of edges $\{v_i, v_{i+1}\} \in E(G)$. A walk is a *path* if all v_i are distinct. If for such a path with $k \geq 2$, $\{v_0, v_k\}$ is also an edge in G, then $v_0, v_1, \ldots, v_k, v_0$ is a *cycle.*

Remark 1.30. The above definition treats paths and cycles as having a start point and an endpoint (so reversing a path technically gives a different path). However, in practice, when we use language like "there is a unique path between u and v", we mean that the path or cycle is unique up to this choice.

Definition 1.31. The *length* of a path, cycle or walk is the number of edges in it.

Example 1.32.



$v_5v_1v_3v_4 \equiv$ path of length 3;

$v_1v_2v_3v_1 \equiv$ cycle of length 3;

Can get a sequence with the same starting and ending points by rotation.

$v_5v_1v_2v_3v_1v_6 \equiv$ walk of length 5.

↳ just a walk

doesn't distinct ⇒ not a path

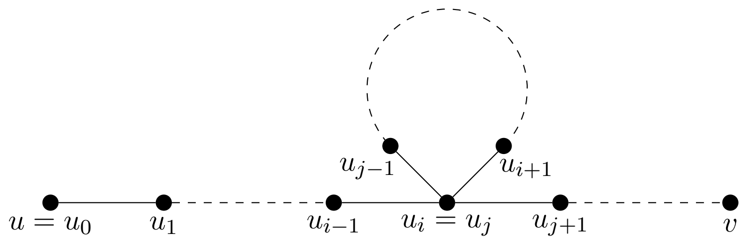
※ path ⊃ walk

by cut down all cycles.

Proposition 1.33. *Every walk from u to v in G contains a path between u and v.*

Proof. By induction on the length ℓ of the walk $u = u_0, u_1, \ldots, v_\ell = v$.

If $\ell = 1$ then our walk is just an edge, which is also a path. Otherwise, if our walk is not a path there is $u_i = u_j$ with $i < j$, then $u = u_0, \ldots, u_i, u_{j+1}, \ldots, v$ is a shorter walk from u to v. We can use induction to conclude that the walk contains a path.



□

In graph theory, we typically ask questions of the following form: what conditions can we impose on a graph to force it to contain certain subgraphs? The following result is an example of this kind of problem.

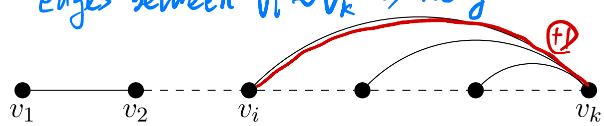
Proposition 1.34. *Every G with minimum degree $\delta(G) = \delta \geq 2$ contains a path of length δ and a cycle of length at least $\delta + 1$.*

$k-1 \geq$ all v_k's neighbour $= d(v_k) \geq \delta$

Proof. Let $v_1, \ldots, v_k$ be a longest path in G. Then all neighbours of v_k belong to $v_1, \ldots, v_{k-1}$, as otherwise we could extend the path. So $k - 1 \geq \delta$ and $k \geq \delta + 1$, and our path has at least δ edges. Let i $(1 \leq i \leq k-1)$ be the minimum index such that $(v_i, v_k) \in E(G)$. Then the neighbours of v_k are among $v_i, \ldots, v_{k-1}$, so $k - i \geq \delta$. Then $v_i, v_{i+1}, \ldots, v_k$ is a cycle of length at least $\delta + 1$.

edges between $v_i \sim v_k \geq$ neighbour of $v_k \geq \delta$

$k-i+1 \geq \delta+1$



□

solved!

Remark 1.35. Note that we have also proved that a graph with minimum degree $\delta \geq 2$ contains cycles of at least $\delta - 1$ different lengths. This fact, and the statement of Proposition 1.34, are both best possible; to see this, consider the complete graph $G = K_{\delta+1}$.

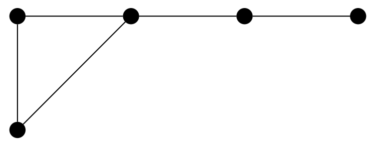
1.8 Connectivity

Paths allow us to travel from one endpoint to another along the edges of the graph. The concept of connectivity concerns whether such paths are guaranteed to exist.

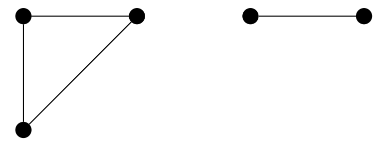
Definition 1.36. A graph G is *connected* if for all pairs $u, v \in V(G)$, there is a path in G from u to v.

Note that it suffices for there to be a walk from u to v, by Proposition 1.33.

Example 1.37.



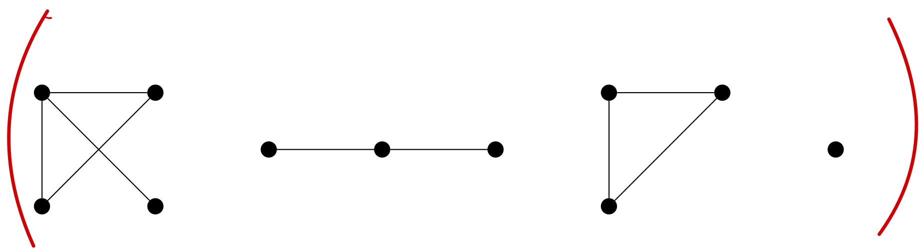
connected



not connected

Definition 1.38. A *(connected) component* of G is a connected subgraph that is maximal by inclusion. G is *connected* if and only if it has one connected component.

Example 1.39.

$G =$ 

has 4 connected components.

what if there are 2 edges between $\{v_i, v_j\}$?

Proposition 1.40. *A graph G with n vertices and m edges has at least $n-m$ connected components.*

减最多的状况

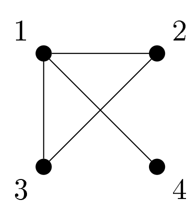
Proof. We start with the empty graph on n vertices, and form G by adding its edges one-by-one, keeping track of the number of components during the process. At the beginning, in the empty graph, each individual vertex is a connected component, so we have n components.

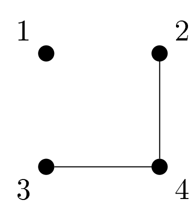
Now, if we add the edge $e = \{u, v\}$, we merge the components of u and v, and so each edge can decrease the number of components by at most one. Hence, after adding all m edges, we have at least $n - m$ connected components. $\square$

1.9 Graph operations and parameters

Definition 1.41. Given $G = (V, E)$, the *complement* $\overline{G}$ of G has the same vertex set V and $(u, v) \in E(\overline{G})$ if and only if $\{u, v\} \notin E(G)$.

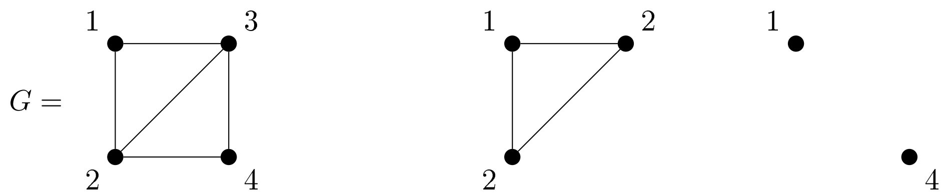
Example 1.42.

$G =$ 

$\overline{G} =$ 

Definition 1.43. A *clique* in G is a complete subgraph in G. An *independent set* is an empty induced subgraph in G.

Example 1.44.



clique in G independent set in G

Definition 1.45. Let $G = (V, E)$ be a graph. The *clique number* $\omega(G)$ denotes the maximum number of vertices in a clique in G. The *independence number* let $\alpha(G)$ denote the number of vertices in a maximum-size independent set in G.

Example 1.46. In Example 1.44, $\omega(G) = 3$ and $\alpha(G) = 2$.

Claim 1.47. A vertex set $U \subseteq V(G)$ is a clique if and only if $U \subseteq V\left(\overline{G}\right)$ is an independent set.

Corollary 1.48. *We have* $\omega(G) = \alpha\left(\overline{G}\right)$ *and* $\alpha(G) = \omega\left(\overline{G}\right)$.

2 Trees

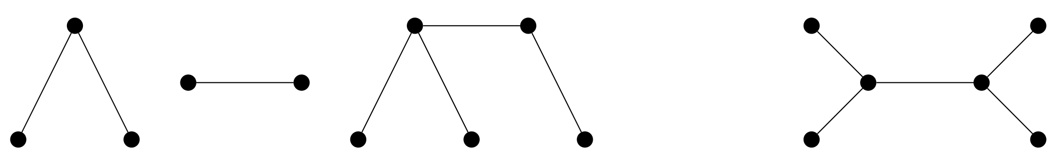
As previously mentioned, in several applications, one of the key properties of a graph is that it should be connected — that is, one should be able to travel from any one vertex to any other. In this chapter, we will take a closer look at connected graphs with as few edges as possible.

2.1 Trees

We begin with some central definitions, which are depicted in the examples below.

Definition 2.1. A graph having no cycle is *acyclic*, and an acyclic graph is called a *forest*. A *tree* is a connected acyclic graph, and a *leaf* (or *pendant vertex*) is a vertex of degree 1.

Example 2.2.

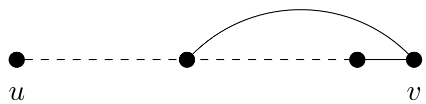


A forest A tree

Our first result shows that every tree must contain leaves.

Lemma 2.3. *Every finite tree with at least two vertices has at least two leaves. Furthermore, deleting a leaf from an n-vertex tree produces a tree with $n-1$ vertices.*

Proof. Let P be a maximal path (by inclusion) in the tree. Since the tree is connected and has at least two vertices, P must have positive length. Let $v \in V(P)$ be an endpoint of the path. By the maximality of P, the neighbours of v must all lie on the path P, as otherwise we could extend P to a longer path. If v had two or more neighbours on P, then we would obtain a cycle, contradicting the tree being acyclic. Hence, v must have degree one. Thus, the two endpoints of P provide the two desired leaves.



(If v had multiple neighbours on the path we would get a cycle).

Suppose v is a leaf of a tree G, and let $G' = G \backslash v$, the subgraph obtained by deleting v and its incident edge. If $u, w \in V(G')$, then no u, w-path P in G can pass through the vertex v of degree 1, so P is also present in G'. Hence, G' is connected. Since deleting a vertex cannot create a cycle, G' is also acyclic. We conclude that G' is a tree with $n-1$ vertices. □

The last part of Lemma 2.3 is particularly important, as it allows us to prove statements about trees by induction on the number of vertices.

2.2 Equivalent definitions of trees

We defined trees as connected acyclic graphs. The following theorem provides some alternative descriptions of trees.

Theorem 2.4. *For an n-vertex simple graph G (with $n \geq 1$), the following are equivalent (and characterise the trees with n vertices).*

(a) G is connected and has no cycles.

(b) G is connected and has $n-1$ edges.

(c) G has $n-1$ edges and no cycles.

(d) For every pair $u, v \in V(G)$, there is exactly one u, v-path in G.

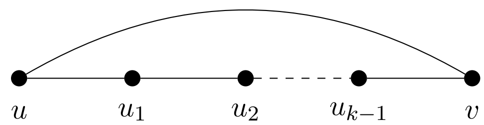
To prove this theorem, we will need a small lemma.

Definition 2.5. An edge of a graph is a *cut-edge* if its deletion disconnects the graph.

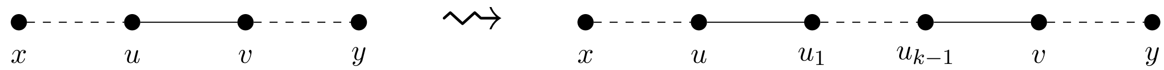
⇕ the edge whose deletion increases the number of components.

Lemma 2.6. *An edge contained in a cycle is not a cut-edge.*

Proof. Let $\{u, v\}$ belong to a cycle.



Then any path $x \dots y$ in G which uses the edge $\{u, v\}$ can be extended to a walk in $G \setminus \{u, v\}$ by using the rest of the cycle, as shown below:



By Proposition 1.33, there is an x, y-path within this walk, and so we deduce that $G \setminus \{u, v\}$ is still connected. $\square$

With this lemma, we can now prove the theorem, showing that the different characterisations of trees are all equivalent.

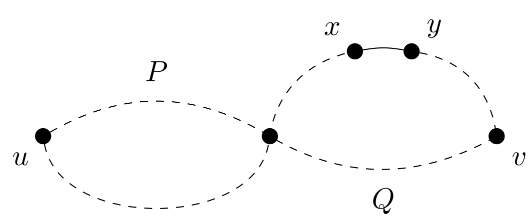
Proof of Theorem 2.4. We first demonstrate the equivalence of (a), (b) and (c) by proving that any two of {connected, acyclic, $n - 1$ edges} implies the third.

(a) $\implies$ (b), (c): We use induction on n. For $n = 1$, an acyclic 1-vertex graph has no edge. For the induction step, suppose $n > 1$, and suppose the implication holds for graphs with fewer than n vertices. Given G, Lemma 2.3 provides a leaf v and states that $G' = G \backslash v$ is acyclic and connected. Applying the induction hypothesis to G' yields $e(G') = n - 2$, and hence, since v is incident to a single edge, $e(G) = n - 1$.

(b) $\implies$ (a), (c): Delete edges from cycles of G one by one until the resulting graph G' is acyclic. By Lemma 2.6, G' is still connected. By the paragraph above, G' has $n - 1$ edges. Since this equals $|E(G)|$, no edges were deleted, and G itself is acyclic.

(c) $\implies$ (a), (b): Suppose G has k components with orders $n_1, \dots, n_k$. Since G has no cycles, each component satisfies property (a), and by the first paragraph the ith component has $n_i - 1$ edges. Summing this over all components yields $e(G) = \sum(n_i - 1) = n - k$. We are given $e(G) = n - 1$, so $k = 1$, and G is connected.

(a) $\implies$ (d): Since G is connected, G has at least one u, v-path for each pair $u, v \in V(G)$. Suppose G has distinct u, v-paths P and Q. Let $e = \{x, y\}$ be an edge in P but not in Q. The concatenation of P with the reverse of Q is a closed walk in which e appears exactly once. Hence, $(P \cup Q) \backslash e$ is an x, y-walk not containing e. By Proposition 1.33, this contains an x, y-path, which completes a cycle with e and contradicts the hypothesis that G is acyclic. Hence G has exactly one u, v-path.



(d) $\implies$ (a): If there is a u, v-path for every $u, v \in V(G)$, then G is connected. If G has a cycle C, then G has two paths between any pair of vertices on C. $\square$

Definition 2.7. Given a connected graph G, a *spanning tree* T is a subgraph of G which is a tree and contains every vertex of G.

This establishes that the different characterisations of trees are indeed equivalent, and hence, for any application, we can choose the most suitable one. We conclude by collecting a few more facts about trees.

Corollary 2.8.

(a) Every connected graph on n vertices has at least $n-1$ edges and contains a spanning tree;

(b) Every edge of a tree is a cut-edge;

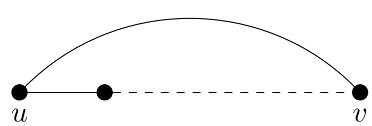
(c) Adding an edge to a tree creates exactly one cycle.

Proof.

(a) Delete edges from cycles of G one by one until the resulting graph G' is acyclic. By Lemma 2.6, G' is connected. The resulting graph is acylic so it is a tree, and contains $n-1$ edges. Therefore G had at least $n-1$ edges and contained a spanning tree.

(b) Note that deleting an edge from a tree T on n vertices leaves $n-2$ edges, so the graph is disconnected by (a).

(c) Let $u, v \in T$. There is a unique path in T between u and v, so adding an edge (u, v) closes this path to a unique cycle.



$\square$

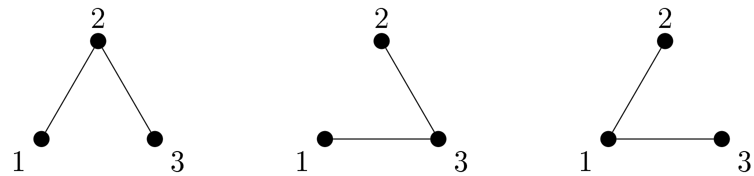
2.3 Cayley's formula – Prüfer code

Now that we know what a tree is, we turn our attention to determining how many there are.

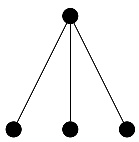
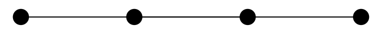
Question 2.9. What is the number of spanning trees in a labelled complete graph on n vertices?

Example 2.10.

$n = 3$:



$n = 4$:

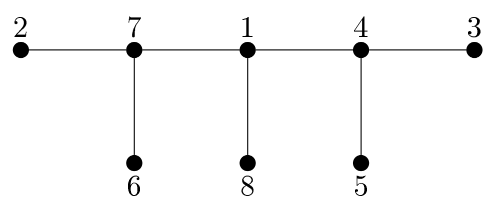
4 stars  $12 = 4!/2$ paths 

Theorem 2.11 (Cayley's Formula)**.** *There are n^{n-2} trees with vertex set $[n]$.*

We shall give two proofs of Cayley's formula. In our first proof, we establish a bijection between trees on $[n]$ and sequences in $[n]^{n-2}$.

Definition 2.12 (Prüfer code)**.** Let T be a tree on an ordered set S of n vertices. To compute the Prüfer sequence $f(T)$, iteratively delete the leaf with the smallest label and append the label of its *neighbour* to the sequence. After $n-2$ iterations a single edge remains and we have produced a sequence $f(T)$ of length $n-2$.

Example 2.13.

$T =$ 

We compute the Prüfer code for T as follows:

- Delete leaf 2, append 7 to the code
- Delete leaf 3, append 4 to the code
- Delete leaf 5, append 4 to the code
- Deleat leaf 4, append 1 to the code
- Delete leaf 6, append 7 to the code
- Deleat leaf 7, append 1 to the code

The edge remaining is $(1, 8)$. We then have $f(T) = (7, 4, 4, 1, 7, 1)$.

Proposition 2.14. *For an ordered n-element set S, the Prüfer code f is a bijection between the trees with vertex set S and the sequences in S^{n-2}.*

n-2

Proof. We need to show every sequence $(a_1, \dots, a_{n-2}) \in S^{n-2}$ defines a unique tree T such that $f(T) = (a_1, \dots, a_{n-2})$. If $n = 2$, then there is exactly one tree on 2 vertices and the algorithm defining f always outputs the empty sequence, the only sequence of length zero. So the claim clearly holds for $n = 2$.

Now, assume $n > 2$ and the claim holds for all ordered vertex sets S' of size less than n. Consider a sequence $(a_1, \dots, a_{n-2}) \in S^{n-2}$. We need to show that $(a_1, \dots, a_{n-2})$ can be uniquely produced by the algorithm.

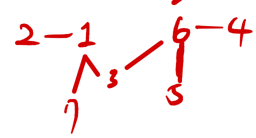
Suppose that the algorithm produces $f(T) = (a_1, \dots, a_{n-2})$ for some tree T. Then the vertices $\{a_1, \dots, a_{n-2}\}$ are precisely those that are not a leaf in T. Indeed, if a vertex v is a leaf in T then it can only appear in $f(T)$ if its neighbour gets deleted during the algorithm. But this would leave v as an isolated vertex, which is impossible. Conversely, if a vertex v is not a leaf then one of its neighbours must be deleted during the algorithm (it cannot be itself deleted before this happens). When this neighbour of v is deleted, v will be added to the Prüfer code for T, so is in $\{a_1, \dots, a_{n-2}\}$.

This implies that the label of the first leaf removed from T is the minimum element of the set $S \backslash \{a_1, \dots, a_{n-2}\}$. Let v be this element. In other words, in every tree T such that $f(T) = (a_1, \dots, a_{n-2})$ the vertex v is a leaf whose unique neighbour is a_1.

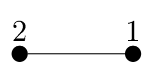
By induction, there is a unique tree T' with vertex set $S \backslash v$ such that $f(T') = (a_2, \dots, a_{n-2})$. Adding the vertex v and the edge (a_1, v) to T' yields the desired unique tree T with $f(T) = (a_1, \dots, a_{n-2})$. $\square$

必非叶

Example 2.15. We use the idea of the above proof to compute the tree with Prüfer code (1,6,6,3,1).



(1,6,6,3,1)
2 is the smallest leaf,
incident to $a_1 = 1$

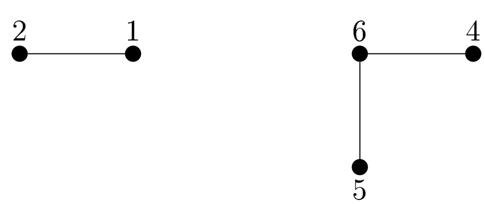


$\{1, 3, 4, 5, 6, 7\}$ left

(6,6,3,1)
4 is the smallest leaf
incident to 6

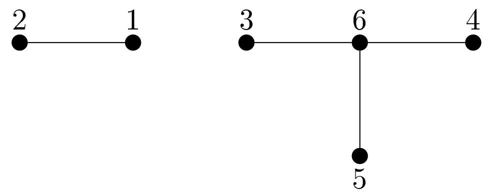
$\{1, 3, 5, 6, 7\}$ left

(6,3,1)
5 is the smallest leaf,
incident to 6



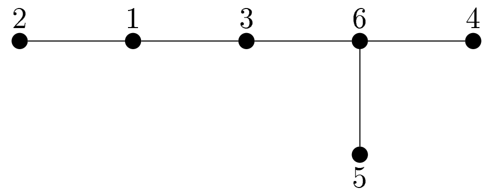
$\{1, 3, 6, 7\}$ left

(3,1)
6 is the smallest leaf
incident to 3



$\{1, 3, 7\}$ left

(1)
3 is the smallest leaf
incident to 1

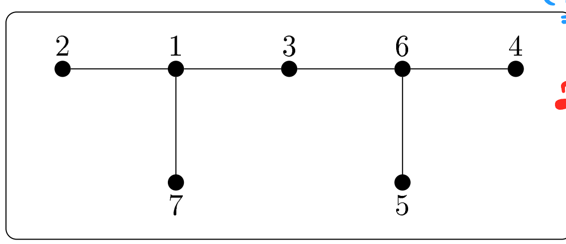


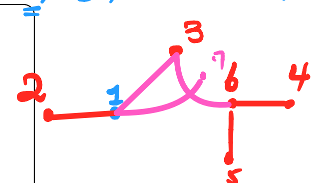
$\{1, 7\}$ left

3, 4, 5, 7   7 : 1 ~ 7

(0, 0, 0, 0, 1)

Now add an edge between
the remaining vertices $\{1, 7\}$.





1 3 7 6

To prove Cayley's formula, just apply Proposition 2.14 with the vertex set $[n]$ (note that there are n^{n-2} sequences in $[n]^{n-2}$).

2.4 Cayley's formula — directed graphs

For our second proof of Cayley's formula we need the following definition.

Definition 2.16. A *directed graph*, or *digraph* for short, is a vertex set and an edge (multi-)set of *ordered* pairs of vertices. Equivalently, a digraph is a (possibly not simple) graph where each edge is assigned a direction. The *out-degree* (respectively *in-degree*) of a vertex is the number of edges incident to that vertex which point away from it (respectively, towards it).

Proof of Cayley's formula (due to Joyal, 1981). We count trees on n vertices which have two distinguished vertices called the "left end" L and the "right end" R, where L and R can be the same vertex. Let t_n be the number of labelled trees on n vertices, and let T_n be the family of labelled trees with two distinguished vertices L and R. Clearly, $|T_n| = t_n n^2$, and it is thus enough to prove that $|T_n| = n^n$. We'll describe a bijection between T_n and the set of all mappings $f : [n] \to [n]$. As the number of such mappings is clearly n^n, the result will follow.

有向樹

So, let $f : [n] \to [n]$ be a mapping. We represent f as a directed graph G_f with vertex set $[n]$ and the set of directed edges $E(G_f) = \{(i, f(i)) : 1 \le i \le n\}$.

好樹

Example.

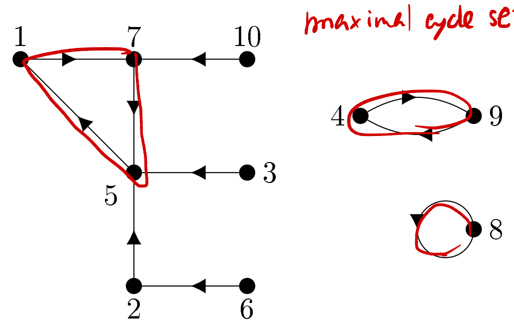
$$f = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 7 & 5 & 5 & 9 & 1 & 2 & 5 & 8 & 4 & 7 \end{pmatrix},$$

n ~ n   1 ~ n ...

$\Rightarrow n^n$ 種

$f_M = \begin{matrix} 1 & 4 & 5 & 7 & 8 & 9 \\ 7 & 9 & & 5 & 8 & 4 \end{matrix}$

maximal cycle set

$G_f =$ 

Observe that G_f is a digraph in which the outdegree of every vertex is exactly one ($f(i)$ is the only out-neighbour of i).

Let us look at a component of G_f (ignoring edge directions for a moment). Since the out-degree of every vertex is exactly one, each such component contains as many edges as vertices ? and has therefore exactly one cycle (by Corollary 2.8). This is easily seen to be a directed cycle (just follow an edge leaving a current vertex until you hit a previously visited vertex).

Let M be the union of the vertex sets of these cycles. In order to create a tree, we need to get rid of these cycles. It is easy to see that f restricted to M is a bijection; moreover, M is the unique maximal set on which f acts as a bijection.

Let us write

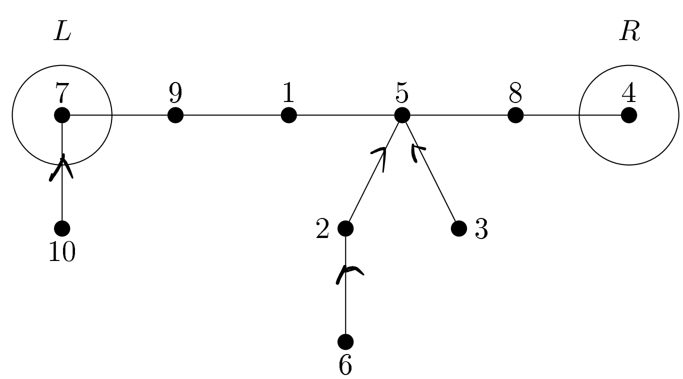
$$f_M = \begin{pmatrix} v_1 & \dots & v_k \\ f(v_1) & \dots & f(v_k) \end{pmatrix},$$

where $v_1 < v_2 < \cdots < v_k$ (and $M = \{v_1, v_2, \dots, v_k\}$). This gives us the ordered k-tuple $(f(v_1), \dots, f(v_k))$. Now we can choose $L = f(v_1)$, $R = f(v_k)$. The tree T corresponding to f is constructed as follows: Draw a (directed) path $f(v_1), f(v_2), \dots, f(v_k)$, and fill in the remaining vertices as in G_f (removing edge directions).

Example (continued).

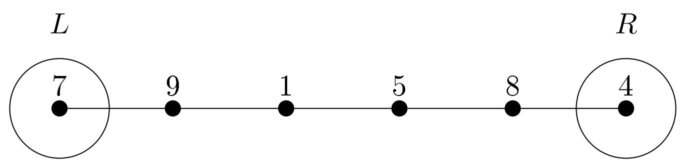
☆ Cycle in directed graph would contains all out-deg of vertices

$$M = \{1, 4, 5, 7, 8, 9\},$$
$$f|_M = \begin{pmatrix} 1 & 4 & 5 & 7 & 8 & 9 \\ 7 & 9 & 1 & 5 & 8 & 4 \end{pmatrix},$$

$T = T(f) =$ 

Reversing the correspondence is easy: given a tree T with two special vertices L and R, look at the unique path P of T connecting L and R. The vertices of P form the set M. Ordering the vertices of M gives us the first line of $f|_M$, the second line is given by the order of the vertices in P, from L to R.

Example (continued)**.**

$P =$ 

$$M = \{1, 4, 5, 7, 8, 9\},$$
$$f|_M = \begin{pmatrix} 1 & 4 & 5 & 7 & 8 & 9 \\ 7 & 9 & 1 & 5 & 8 & 4 \end{pmatrix}.$$

The remaining values of f are then filled in accordance with the unique paths from the remaining vertices to P (directing these paths towards P). □

3 Connectivity

While trees are connected graphs, their connectivity is very fragile — every edge is a cut-edge, so if any edge is removed, the graph is no longer connected. For many applications it is important to have graphs with more robust connectivity, and in this chapter we explore how this can be quantified.

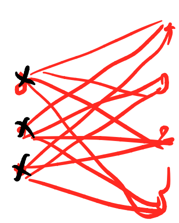
3.1 Vertex connectivity

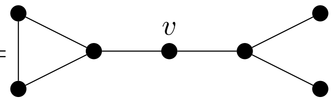
In this section we explore the question of how many vertices need to be removed to disconnect a graph.

Definition 3.1. A *vertex cut* in a connected graph $G = (V, E)$ is a set $S \subseteq V$ such that $G \backslash S := G[V \backslash S]$ has more than one connected component. A *cut vertex* is a vertex v such that $\{v\}$ is a cut.

Definition 3.2. G is called k*-connected* if $|V(G)| > k$ and if $G \backslash X$ is connected for every set $X \subseteq V$ with $|X| < k$. In other words, no two vertices of G are separated by fewer than k other vertices. Every (non-empty) graph is 0-connected and the 1-connected graphs are precisely the non-trivial connected graphs. The greatest integer k such that G is k-connected is the *connectivity* $\kappa(G)$ of G.

Example 3.3.
- $G = K_n$: $\kappa(G) = n - 1$



- $G = K_{m,n}$, $m \leq n$: $\kappa(G) = m$. Indeed, let G have bipartition $A \cup B$, with $|A| = m$ and $|B| = n$. Deleting A disconnects the graph. On the other hand, deleting $S \subset V$ with $|S| < m$ leaves both $A \backslash S$ and $B \backslash S$ non-empty and any $a \in A \setminus S$ is connected to any $b \in B \setminus S$. Hence $G \backslash S$ is connected.
- $G =$  : $\kappa(G) = 1$. Deleting v disconnects G, so v is a cut vertex.

Our first bound shows that in order to have high connectivity, you need all vertices to have large degree.

Proposition 3.4. *For every graph G, $\kappa(G) \leq \delta(G)$.*

Proof. If G is a complete graph then trivially $\kappa(G) = \delta(G) = |G| - 1$. Otherwise let $v \in G$ be a vertex of minimum degree $d(v) = \delta(G)$. Deleting $N(v)$ disconnects v from the rest of G. □

Remark 3.5. High minimum degree does not imply connectivity. Consider two disjoint copies of K_n.

$\lfloor \frac{n}{2} \rfloor$

K_n K_n

? However, the converse is *locally* true.

← by the bound

Theorem 3.6 (Mader 1972)**.** *Every graph of average degree at least $4k$ has a k-connected subgraph.*

Proof. For $k \in \{0, 1\}$ the assertion is trivial; we consider $k \geq 2$ and a graph $G = (V, E)$ with $|V| = n$ and $|E| = m$. For inductive reasons it will be easier to prove the stronger assertion that G has a k-connected subgraph whenever

(i) $n \geq 2k - 1$ and

$\geq 2k^2 - 3k$

(ii) $m \geq (2k - 3)(n - k + 1) + 1$. $\geq (2k-3)(k)+1$

(This assertion is indeed stronger, i.e. (i) and (ii) follow from our assumption of $\bar{d}(G) \geq 4k$: (i) holds since $n > \Delta(G) \geq 4k$, while (ii) follows from $m = \frac{1}{2}\bar{d}(G)n \geq 2kn$.)

We apply induction on n. If $n = 2k - 1$, then $k = \frac{1}{2}(n + 1)$, and hence

$$m \geq (n - 2)\frac{n + 1}{2} + 1 = \frac{1}{2}n(n - 1)$$

by (ii). Thus $G = K_n \supseteq K_{k+1}$, proving our claim. We therefore assume that $n \geq 2k$. If v is a vertex with $d(v) \leq 2k - 3$, we can apply the induction hypothesis to $G \backslash v$ and are done. So we assume that $\delta(G) \geq 2k - 2$. If G is itself not k-connected, then there is a separating set $X \subseteq V$ with fewer than k vertices, such that $G \backslash X$ is disconnected. Let V_1 be one component of $G \backslash X$ and let V_2 be the union of the other components. Let $G_i = G[V_i \cup X]$, so that $G = G_1 \cup G_2$, and every edge of G is either in G_1 or G_2 (or both). Each vertex in each V_i has at least $\delta(G) \geq 2k - 2$ neighbours in G and thus also in G_i, so $|G_1|, |G_2| \geq 2k - 1$. Note that each $|G_i| < n$, so by the induction hypothesis, if no G_i has a k-connected subgraph then each

$$e(G_i) \leq (2k - 3)(|G_i| - k + 1).$$

Hence,

$$\begin{aligned} m &\le e(G_1) + e(G_2) \\ &\le (2k-3)(|G_1| + |G_2| - 2k + 2) \\ &\le (2k-3)(n-k+1) \quad (\text{since } |G_1 \cap G_2| \le k-1), \end{aligned}$$

contradicting (ii). $\square$

3.2 Edge connectivity

The previous section dealt with removing vertices from a graph. Here we consider disconnecting a graph by removing edges instead.

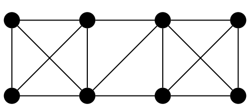
$V \setminus S$

Definition 3.7. A *disconnecting set* of edges is a set $F \subseteq E(G)$ such that $G \setminus F$ has more than one component. Given $S, T \subset V(G)$, the notation $[S, T]$ specifies the set of edges having one endpoint in S and the other in T. An *edge cut* is an edge set of the form $[S, \overline{S}]$, where S is a non-empty proper subset of $V(G)$. A graph is *k-edge-connected* if every disconnecting set has at least k edges. The *edge-connectivity* of G, written $\kappa'(G)$, is the minimum size of a disconnecting set. A single edge that disconnects G is called a *bridge* (or a *cut-edge*).

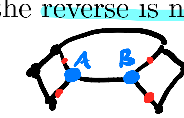
$|[X, \overline{X}]| \ge k$

任取(k-1)邊拔皆連通
$\Updownarrow$
至少任取k邊斷都不連通

Example 3.8.

- $G = K_n$: $\kappa'(G) = n - 1$.
- $G =$  : $\kappa'(G) = 3$, whereas $\kappa(G) = 2$.

Remark 3.9. An edge cut is a disconnecting set but the reverse is not necessarily true. However, every minimal disconnecting set is a cut.



$[AB, \overline{AB}]$ = ·

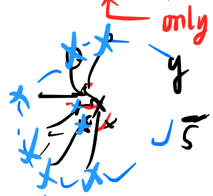
Theorem 3.10. $\kappa(G) \le \kappa'(G) \le \delta(G)$.

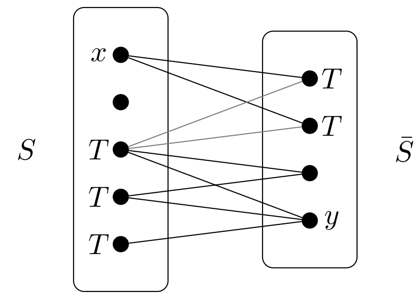
Proof. The edges incident to a vertex v of minimum degree form a disconnecting set; hence $\kappa'(G) \le \delta(G)$. It remains to show $\kappa(G) \le \kappa'(G)$. Suppose $|G| > 1$ and $[S, \overline{S}]$ is a minimum edge cut, having size $\kappa'(G)$.

If every vertex of S is adjacent to every vertex of $\overline{S}$, then $\kappa'(G) = |S||\overline{S}| = |S|(|G| - |S|)$. This expression is minimized at $|S| = 1$. By definition, $\kappa(G) \le |G| - 1$, so the inequality holds.

Hence we may assume there exists $x \in S$, $y \in \overline{S}$ with x not adjacent to y. Let T be the vertex set consisting of all neighbours of x in $\overline{S}$ and all vertices of $S \setminus x$ that have neighbours in $\overline{S}$ (illustrated below). Deleting T destroys all the edges in the cut $[S, \overline{S}]$ (but does not delete x or y), so T is a separating set. Now, by the definition of T we can injectively ? associate at least one edge of $[S, \overline{S}]$ to each vertex in T, so $\kappa(G) \le |T| \le \left|[S, \overline{S}]\right| = \kappa'(G)$.

only deleting vertices, but $S \setminus x$ who have neighbor in $\overline{S}$ may have 1^+ neighbor in $\overline{S}$.





□

3.3 2-connected graphs

Definition 3.11. Two paths are *internally disjoint* if neither contains a non-endpoint vertex of the other. We denote the length of the shortest path from u to v (the *distance* from u to v) by $d(u, v)$.

distance

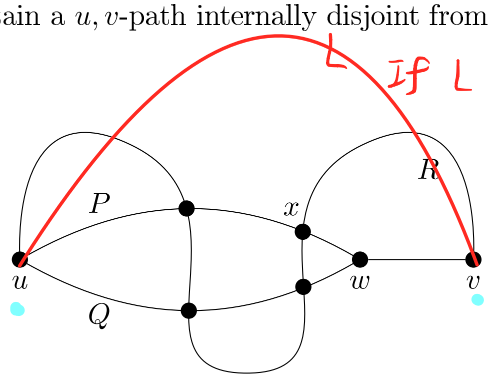
Theorem 3.12 (Whitney 1932)**.** *A graph G having at least three vertices is 2-connected if and only if each pair $u, v \in V(G)$ is connected by a pair of internally disjoint u, v-paths in G.*

Proof. When G has internally disjoint u, v-paths, deletion of one vertex cannot separate u from v. Since this is given for every u, v, the condition is sufficient. For the converse, suppose that G is 2-connected. We prove by induction on $d(u, v)$ that G has two internally disjoint u, v paths. When $d(u, v) = 1$, the graph $G \backslash \{u, v\}$ is connected, since $\kappa'(G) \geq \kappa(G) = 2$. $\boxed{\text{A}}$ u, v-path in $G \backslash \{u, v\}$ is internally disjoint in G from the u, v-path consisting of the edge $\{u, v\}$ itself.

对k也有 internally disjoint path

For the induction step, we consider $d(u, v) = k > 1$ and assume that G has internally disjoint x, y-paths whenever $1 \leq d(x, y) < k$. Let w be the vertex before v on a shortest u, v-path. We have $d(u, w) = k - 1$, and hence by the induction hypothesis G has internally disjoint u, w-paths P and Q. Since $G \backslash w$ is connected, $G \backslash w$ contains a u, v-path R. If this path avoids P or Q, we are finished, but R may share internal vertices with both P and Q. Let x be the last vertex of R belonging to $P \cup Q$. Without loss of generality, we may assume $x \in P$. We combine the u, x-subpath of P with the x, v-subpath of R to obtain a u, v-path internally disjoint from $Q \cup \{(w, v)\}$.

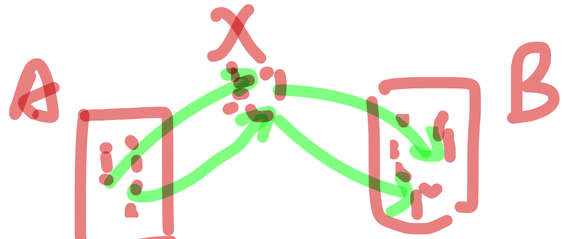
If L avoids P and Q



by hypothesis, we have P, Q paths is the core!

□

Corollary 3.13. *G is 2-connected and $|G| \geq 3$ if and only if every two vertices in G lie on a common cycle.*



3.4 Menger's Theorem

Definition 3.14. Let $A, B \subseteq V$. An A-B path is a path with one endpoint in A, the other endpoint in B, and all interior vertices outside of $A \cup B$. Any vertex in $A \cap B$ is a trivial A-B path.

If $X \subseteq V$ (or $X \subseteq E$) is such that every A-B path in G contains a vertex (or an edge) from X, we say that X *separates* the sets A and B in G. This implies in particular that $A \cap B \subseteq X$.

Theorem 3.15 (Menger 1927). *Let $G = (V, E)$ be a graph and let $S, T \subseteq V$. Then the maximum number of vertex-disjoint S-T paths is equal to the minimum size of an S-T separating vertex set.*

Proof. Obviously, the maximum number of disjoint paths does not exceed the minimum size of a separating set, because for any collection of disjoint paths, any separating set must contain a distinct vertex from each path. So we just need to prove there is an S-T separating set and a collection of disjoint S-T paths with the same size.

We use induction on $|E|$, the case $E = \varnothing$ being trivial. We first consider the case where S and T are disjoint.

Let k be the minimum size of an S-T separating vertex set. Choose $e = \{u, v\} \in E$. Let $G' = (V, E \setminus e)$. If each S-T separating vertex set in G' has size at least k, then inductively there exist k vertex-disjoint S-T paths in G', hence in G.

So we can assume that G' has an S-T separating vertex set C of size at most $k-1$. Then $C \cup \{u\}$ and $C \cup \{v\}$ are S-T separating vertex sets of G of size k.

Since C is a separating set for G', no component of $G' \backslash C$ has elements from both S and T. Let V_S be the union of components with elements from S, and let V_T be the union of components with elements in T. If we were to add the edge (u, v) to $G' \backslash C$ then there would be a path from S to T (because C does not separate S and T in G). So, without loss of generality $u \in V_S$ and $v \in V_T$.

Now, each S-$(C \cup \{u\})$ separating vertex set B of G' has size at least k, as it is S-T separating in G. Indeed, each S-T path P in G intersects $C \cup \{u\}$. Let P' be the subpath of P that goes from S to the first time it touches $C \cup \{u\}$. If P' ends with a vertex in C, then $u \notin P'$ so P' is an S-$(C \cup \{u\})$ path in G'. If P' ends in u, then it is disjoint from C and so by the above it contains only vertices in V_S. So $v \notin P'$ and again P' is an S-$(C \cup \{u\})$ path in G'. In both cases we showed that P' is an S-$(C \cup \{u\})$ path in G' so P intersects B.

So by induction, G' contains k disjoint S-$(C \cup \{u\})$ paths. Similarly, G' contains k disjoint $(C \cup \{v\})$-T paths. Any path in the first collection intersects any path in the second collection only in C, since otherwise G' contains an S-T path avoiding C.

Hence, as $|C| = k - 1$, we can pairwise concatenate these paths to obtain $k - 1$ disjoint S-T paths. We can finally obtain a kth path by inserting the e between the path ending at u and the path starting at v.

It remains to consider the general situation where S and T might not be disjoint. Let $X = S \cap T$ and apply the theorem with the disjoint sets $S' = S \backslash X$ and $T' = T \backslash X$, in the graph $G' = G \backslash X$. Let k' be the size of a minimum separating set in G'. We can obtain a $(k' + |X|)$-vertex S-T separating set in G by adding every vertex in X to an S'-T' separating set in G'. Similarly we can obtain a collection of $k' + |X|$ vertex-disjoint S-T paths by adding each vertex in X as a trivial path to a collection of vertex-disjoint S'-T' paths in G'. $\square$

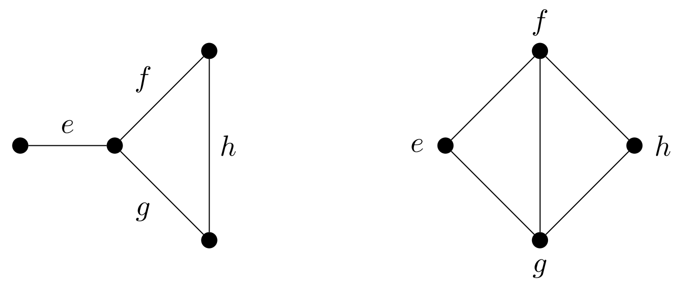
• every cycle in the L(G) correspond to either
a star
or a cycle in G

Corollary 3.16. *For $S \subseteq V$ and $v \in V \setminus S$, the minimum number of vertices distinct from v separating v from S in G is equal to the maximum number of paths forming an v-S fan in G. (that is, the maximum number of $\{v\}$-S paths which are disjoint except at v).*

Proof. Apply Menger's Theorem with $T = N(v)$. Note that none of the resulting paths go through v; if one did, then it would contain two vertices of T, violating the definition of an S-T path. So we have a suitable number of vertex-disjoint S-T paths not including v, and we can append v to each path to give a v-S fan. $\square$

Definition 3.17. The *line graph* of G, written $L(G)$, is the graph whose vertices are the edges of G, with $\{e, f\} \in E(L(G))$ when $e = \{u, v\}$ and $f = \{v, w\}$ in G (i.e. when e and f share a vertex).

Example 3.18.



G $\qquad$ $L(G)$

f, e, g → f. g

Note that a path in $L(G)$ corresponds to a sequence of distinct edges $e_0, \ldots, e_\ell$ in G such that every pair of consecutive edges is incident. This may not be a path in G, because the endpoints might not "line up": it might be that for some i the common endpoint of e_{i-1} and e_i is the same as the common endpoint of e_i and e_{i+1}. But we can always delete a few edges to obtain a path in G: if there exists i as described, we can simply delete the edge e_i, and repeatedly make such deletions until no conflicts remain.

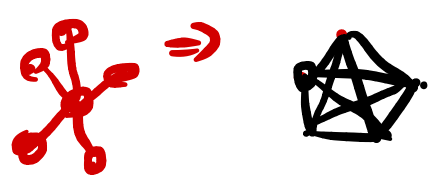
Corollary 3.19. *Let u and v be two distinct vertices of G.*

? 1. *If $\{u, v\}$? $\notin E$, then the minimum number of vertices different from u, v separating u from v in G is equal to the maximum number of internally vertex-disjoint u-v paths in G.*
2. *The minimum number of edges separating u from v in G is equal to the maximum number of edge-disjoint u-v paths in G.*

Proof. For (i), Apply Menger's Theorem with $S = N(u)$ and $T = N(v)$.

For (ii), Apply Menger's Theorem to the line graph of G, with S as the set of edges adjacent to u and T as the set of edges adjacent to v. $\square$

Theorem 3.20 (Global Version of Menger's Theorem).



1. *A graph is k-connected if and only if it contains k internally vertex-disjoint paths between any two vertices.*

2. *A graph is k-edge-connected if and only if it contains k edge-disjoint paths between any two vertices.*

Proof. First we prove (i). if a graph G contains k internally disjoint paths between any two vertices, then $|G| > k$ and G cannot be separated by fewer than k vertices; thus, G is k-connected.

Conversely, suppose that G is k-connected (and, in particular, has more than k vertices) but contains vertices u, v not linked by k internally disjoint paths. By Corollary 3.19, u and v are adjacent; let $G' = G \backslash \{u, v\}$. Then G' contains at most $k-2$ internally disjoint u, v-paths. By Corollary 3.19, we can separate u and v in G' by a set X of at most $k-2$ vertices. As $|G| > k$, there is at least one further vertex $w \notin X \cup \{u, v\}$ in G. Now X separates w in G' from either u or v (say, from u). But then $X \cup \{v\}$ is a set of at most $k-1$ vertices separating w from u in G, contradicting the k-connectedness of G.

Then, (ii) follows straight from Corollary 3.19. $\square$

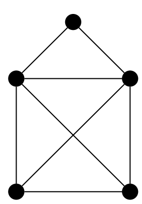
4 Eulerian and Hamiltonian cycles

With connectivity, we were concerned with how one could travel through a graph from one vertex to another. However, sometimes in graph theory, as in life, the journey matters more than the destination, and in this chapter, we shall be more concerned with the vertices and edges we visit along the way.

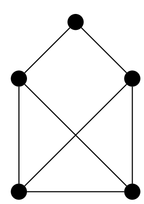
4.1 Eulerian trails and tours

When it comes to Eulerian trails, the question at hand can be (loosely) formulated in the following fashion.

Question 4.1. Which of the two pictures below can be drawn in one go without lifting your pen from the paper?



or

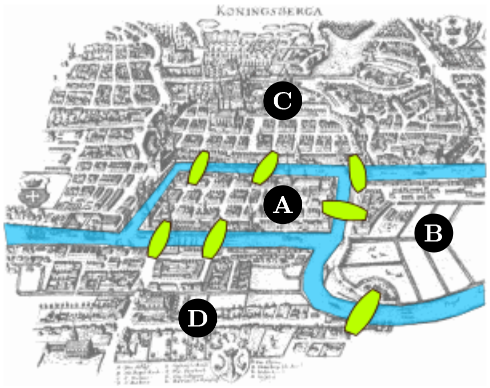


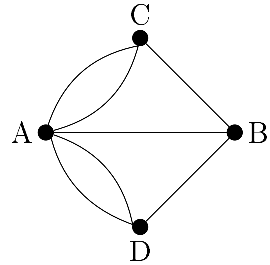
Definition 4.2. A *trail* is a walk with no repeated edges.

Definition 4.3. An *Eulerian trail* in a (multi)graph $G = (V, E)$ is a walk in G passing through every edge exactly once. If this walk is closed (starts and ends at the same vertex) it is called an *Eulerian tour*.

One motivation for this concept is the "Seven bridges of Königsberg" problem:

Question 4.4. Is it possible to design a closed walk passing through all the 7 bridges exactly once? Equivalently, does the graph on the right have an Eulerian tour?





Theorem 4.5. *A connected (multi)graph has an Eulerian tour if and only if each vertex has even degree.*

In order to prove this theorem we use the following lemma.

2- edge-connected

Lemma 4.6. *Every maximal trail in an even graph (i.e., a graph where all the vertices have even degree) is a closed trail.*

Proof. Let T be a maximal trail. If T is not closed, then T has an odd number of edges incident to the final vertex v. However, as v has even degree, there is an edge incident to v that is not in T. This edge can be used to extend T to a longer trail, contradicting the maximality of T. $\square$

$\Rightarrow$

Proof of Theorem 4.5. To see that the condition is necessary, suppose G has an Eulerian tour C. If a vertex v was visited k times in the tour C, then each visit used 2 edges incident to v (one incoming edge and one outgoing edge). Thus, $d(v) = 2k$, which is even.

even graph.

To see that the condition is sufficient, let G be a connected graph with even degrees. Let $T = e_1e_2 \dots e_\ell$ (where $e_i = \{v_{i-1}, v_i\}$) be a longest trail in G. Then, by Lemma 4.6, T is closed, i.e., $v_0 = v_\ell$. If T does not include all the edges of G then, since G is connected, there is an edge e outside of T such that $e = \{u, v_i\}$ for some vertex v_i in T. But then $T' = ee_{i+1} \dots e_\ell e_1 e_2 \dots e_i$ is a trail in G which is longer than T, contradicting the fact that T is a longest trail in G. Thus, we conclude that T includes all the edges of G and so it is an Eulerian tour. $\square$

That is if T is not a trail in an even graph, we can extend it.

Corollary 4.7. *A connected multigraph G has an Eulerian trail if and only if it has either 0 or 2 vertices of odd degree.*

Notice that Theorem 4.5 is $\Leftrightarrow$.

Proof. Suppose T is an Eulerian trail from vertex u to vertex v. If $u = v$ then T is an Eulerian tour and so by Theorem 4.5 it follows that all the vertices in G have even degree. If $u \neq v$, note that the multigraph $G \cup \{e\}$, where $e = \{u, v\}$ is a new edge, has an Eulerian tour, namely $T \cup \{e\}$. By Theorem 4.5 it follows that all the degrees in $G \cup \{e\}$ are even. Thus, we conclude that, in the original multigraph G, the vertices u, v are the only ones which have odd degree.

Now we prove the other direction of the corollary. If G has no vertices of odd degree then by Theorem 4.5 it contains an Eulerian tour which is also an Eulerian trail. Suppose now that G has two vertices u, v of odd degree. Then $G \cup \{e\}$, where $e = \{u, v\}$ is a new edge, only has vertices of even degree and so, by Theorem 4.5, it has an Eulerian tour C. Removing the edge e from C gives an Eulerian trail of G from u to v. $\square$

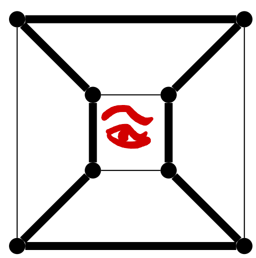
4.2 Hamilton paths and cycles

The previous results show that we understand precisely which graphs admit Eulerian tours. Perhaps surprisingly, it is much harder to determine whether or not we can visit every vertex, instead of edge, of a graph exactly once. For this problem, we will only be able to give a necessary condition and a sufficient condition, but they are far from a full characterisation.

Definition 4.8. A Hamilton path/cycle in a graph G is a path/cycle visiting every vertex of G exactly once. A graph G is called Hamiltonian if it contains a Hamilton cycle.

Hamilton cycles were introduced by Kirkman in 1885, and were named after Sir William Hamilton, who produced a puzzle whose goal was to find a Hamilton cycle in a specific graph.

Example 4.9. Hamilton cycle in the skeleton of the 3-dimensional cube.



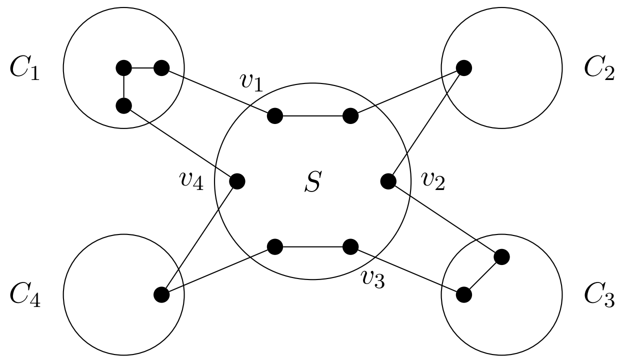
最多連出|S|個

We give some necessary conditions for Hamiltonicity.

connected

Proposition 4.10. *If G is Hamiltonian then for any set $S \subseteq V$ the graph $G \backslash S$ has at most $|S|$ connected components.*

Proof. Let $C_1, \ldots, C_k$ be the components of $G \backslash S$. Imagine that we are moving along a Hamilton cycle in some order, vertex-by-vertex (in the picture below, we are moving clockwise, starting from some vertex in C_1, say). We must visit each component of $G \backslash S$ at least once; when we leave C_i for the first time, let v_i be the subsequent vertex visited (which must be in S). Each v_i must be distinct because a cycle cannot intersect itself. Hence, S must have at least as many vertices as the number of connected components of $G \backslash S$.

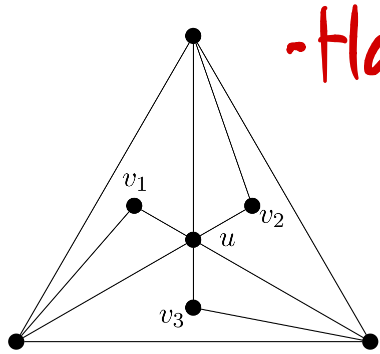


$\square$

Corollary 4.11. *If a connected bipartite graph $G = (V, E)$ with bipartition $V = A \cup B$ is Hamiltonian then $|A| = |B|$.*

Proof. By deleting the vertices in A from G we get $|B|$ isolated vertices and so $G \backslash A$ has $|B|$ connected components. Thus, by Proposition 4.10 we conclude that $|A| \geq |B|$. By symmetry we can also show that $|B| \geq |A|$. Thus, we conclude that $|A| = |B|$. $\square$

Example 4.12. The condition in Proposition 4.10 is not sufficient to ensure that a graph is Hamiltonian. The graph G on the right satisfies the condition of Proposition 4.10 but is not Hamiltonian. Indeed, one would need to include all the edges incident to the vertices v_1, v_2 and v_3 in a Hamilton cycle of G; however, in that case the vertex u would have degree at least 3 in that Hamilton cycle, which is impossible.



-Hamil ⇒ $G \backslash |S|$ has at most $|S|$ component.

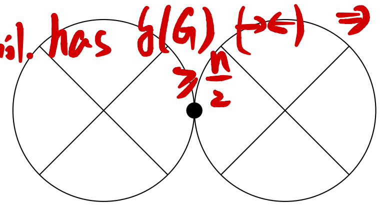
We also give some sufficient conditions for Hamiltonicity.

Theorem 4.13 (Dirac 1952). *If G is a simple graph with $n \geq 3$ vertices and if $\delta(G) \geq n/2$, then G is Hamiltonian.*

Example 4.14. (best-possible minimum degree bound):

- The graph consisting of two cliques of orders $\lfloor (n+1)/2 \rfloor$ and $\lceil (n+1)/2 \rceil$ sharing a vertex has minimum degree $\lfloor (n-1)/2 \rfloor$ but is not Hamiltonian (it is not even 2-connected).

Logic: add edge but keep G non-Hamiltanian
since doesn't ↓ δ(G), let's add it.
If the maximal nonhamil. has δ(G) ≥ n/2 ⇒ δ(G) ≥ n/2 G is Hailtaniam.



9

- If n is odd, then the complete bipartite graph $K_{(n-1)/2,(n+1)/2}$ has minimum degree $\frac{n-1}{2}$ but is not Hamiltonian.

δ(G)

Proof of Theorem 4.13. The condition that $n \geq 3$ must be included since K_2 is not Hamiltonian but satisfies $\delta(K_2) = |K_2|/2$.

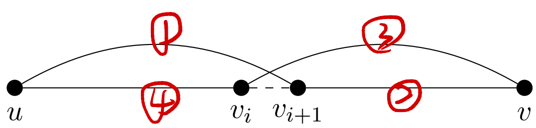
no Hamil. 怎麼加 都不超過 $\frac{n}{2}$

If there is a non-Hamiltonian graph satisfying the hypotheses, then adding edges cannot reduce the minimum degree, so we may restrict our attention to *maximal* non-Hamiltonian graphs G with minimum degree at least $n/2$. By "maximal" we mean that for every pair $\{u, v\}$ of non-adjacent vertices of G, the graph obtained from G by adding the edge $e = \{u, v\}$ is Hamiltonian.

if not, look for the edge connects to the poor vertex.

The maximality of G implies that G has a Hamilton path, say from $u = v_1$ to $v = v_n$, because every Hamilton cycle in $G \cup \{e\}$ must contain the new edge e. We use most of this path $v_1, \ldots, v_n$, with a small switch, to obtain a Hamilton cycle in G. If some neighbour of u immediately follows a neighbour of v on the path, say $\{u, v_{i+1}\} \in E(G)$ and $\{v, v_i\} \in E(G)$, then G has the Hamilton cycle $(u, v_{i+1}, v_{i+2}, \ldots, v_{n-1}, v, v_i, v_{i-1}, \ldots, v_2)$ shown below.

u→v is a Ha path.
contains V(G)



N(u)
N(v)

To prove that such a cycle exists, we show that there is a common index in the sets S and T defined by $S = \{i : \{u, v_{i+1}\} \in E(G)\}$ and $T = \{i : \{v, v_i\} \in E(G)\}$. Summing the sizes of these sets yields

i-1
δ(G) ≥ n/2

(n+1) ONE 無自環

$$|S \cup T| + |S \cap T| = |S| + |T| = d(u) + d(v) \geq n.$$

Neither S nor T contains the index n. This implies that $|S \cup T| < n$, and hence $|S \cap T| \geq 1$, as required. This is a contradiction. □

to G is non-Ha. with δ ≥ n/2

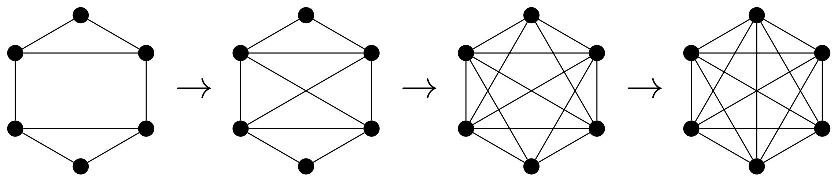
Ore observed that this argument uses only that $d(u) + d(v) \geq n$. Therefore, we can weaken the requirement of minimum degree $n/2$ to require only that $d(u) + d(v) \geq n$ whenever u is not adjacent to v.

Theorem 4.15 (Ore 1960). *If G is a simple graph with $n \geq 3$ vertices such that for every pair of non-adjacent vertices u, v of G we have $d(u) + d(v) \geq |G|$, then G is Hamiltonian.*

4.3 Closures and Chvátal's Condition

Theorem Theorem 4.13 gives a necessary condition for Hamiltonicity, but it is a rather tough requirement — we need the degree of every vertex to be at least $n/2$. In this section we shall provide a different sufficient condition, wherein some vertices are allowed to have lower degrees.

Definition 4.16. The (Hamiltonian) closure of a graph G, denoted $C(G)$, is the supergraph of G on $V(G)$ obtained by iteratively adding edges between pairs of nonadjacent vertices whose degree sum is at least n, until no such pair remains. In the example below, $n = 6$.



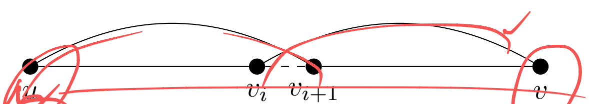
It turns out that regardless of the pairs we choose at each step of creating the closure we always end up with the same final graph, in particular closure of a graph is unique.

Proposition 4.17. *Any graph has a unique closure.*

Proof. Let us assume that $C_1 \neq C_2$ are both closures of a graph G. Let us denote by $e_1, \ldots, e_k$ the edges we added to G in that order to create C_1 and let $f_1, \ldots, f_\ell$ denote the edges we added to G in that order to obtain C_2. Let us assume $|C_1| \leq |C_2|$. Then there must exist an edge $f_i \notin C_1$, let us also assume i is as small as possible. But then at stage i of the second process the current graph was $G' = G \cup \{f_1, \ldots, f_{i-1}\} \subseteq C_1$ and since we added f_i to G' we also must be able to add it to C_1, a contradiction. □

Theorem 4.18 (Bondy-Chvátal, 1976)**.** *A simple n-vertex graph is Hamiltonian if and only if its closure is Hamiltonian.*

Sketch of Proof. Suppose $G \cup \{(u, v)\}$ is Hamiltonian and $d(u) + d(v) \geq n$. Then G has a Hamilton path from u to v. Since $|N(u)| + |N(v)| \geq n$, on the path there is a vertex $v_i \sim v$ such that the next vertex $v_{i+1} \sim u$ (as in the proof of Dirac's Theorem). This gives a Hamilton cycle in G.



□

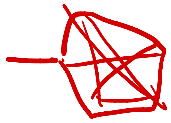
d(u) + d(v) ≥ n

Theorem 4.19 (Chvátal, 1972)**.** *Suppose G has vertex degrees $d_1 \leq \cdots \leq d_n$. If for every $i < n/2$ we have $d_i > i$ or $d_{n-i} \geq n - i$, then G is Hamiltonian.*

Every (d_i, d_{n-i}) $\vee$ i or $\geq$ $n-i$

? If $d_1 \leq 1 \Rightarrow d_1 = 1$ $\Downarrow$ $d_{n-1} \geq n-1$ $\Rightarrow d_1 \to d_{n-1}$

$d_1 = 1$, $d_{n-1} \geq n-1$



②

1, 2, 3

G 1 6 6 6 6 6 7

Proof. Because adding edges to form the closure does not reduce any values in the degree sequence, and a graph is Hamiltonian if and only if its closure is Hamiltonian, it suffices to consider the special case where G is closed. In this case, we prove that the condition implies $G = K_n$. We prove the contrapositive; if $G = C(G) \neq K_n$, then we find some $i \leq n/2$ for which Chvátal's condition is violated. Here violation means that at least i vertices have degree at most i and at least $n - i$ vertices have degree less than $n - i$, implying that $d_i \leq i$ and $d_{n-i} < n - i$.

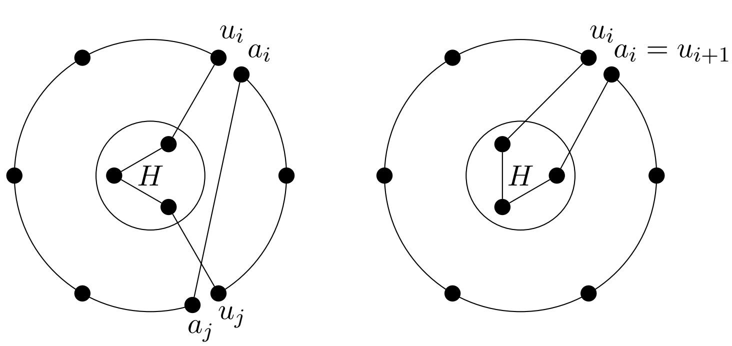
If $G \neq K_n$, then among the pairs of nonadjacent vertices we choose a pair u, v with maximum degree sum. Because G is closed, $u \nsim v$ implies $d(u) + d(v) < n$, and we may assume $d(u) \leq d(v)$. We then have $d(u) < n/2$; let $i = d(u)$.

Because we chose a nonadjacent pair with maximum degree sum, every vertex of $V \backslash \{v\}$ that is not adjacent to v has degree at most $d(u) = i$; furthermore, there are at least $n - 1 - d(v) \geq d(u) = i$ of these vertices. Similarly, every vertex of $V \backslash \{u\}$ that is not adjacent to u has degree at most $d(v) < n - d(u) = n - i$, and there are $n - 1 - d(u) = n - 1 - i$ of these. Since $d(u) \leq d(v)$, we can also add u to the set of vertices with degree at most $d(v)$, so we obtain $n - i$ vertices with degree less than $n - i$. Hence we have proved $d_i \leq i$ and $d_{n-i} < n - i$ for this specially chosen i, which contradicts the hypothesis. □

The final result of this section shows that highly-connected graphs without large independent sets are bound to be Hamiltonian.

Theorem 4.20 (Chvátal-Erdős, 1972)**.** *If $\kappa(G) \geq \alpha(G)$, then G has a Hamiltonian cycle (unless $G = K_2$).*

Proof. The theorem is trivially true for complete graphs. Otherwise, the condition requires $\kappa(G) \geq \alpha(G) > 1$. We will prove that if G has no Hamilton cycle, then $k = \kappa(G) < \alpha(G)$. Let C be a longest cycle (which we are assuming is not Hamiltonian) in G. Let H be a component of $G \backslash C$. Since $\delta(G) \geq \kappa(G)$ and every graph with $\delta(G) \geq 2$ has a cycle of length at least $\delta + 1$ (Proposition 1.34), C has at least $k + 1$ vertices. Also C has at least k vertices with edges to H, else the vertices of C with edges to H contradict $\kappa(G) = k$. Let $u_1, \ldots, u_\ell$ be the vertices of C with edges to H ($\ell \geq k$), in clockwise order. For $i = 1, \ldots, k$, let a_i be the vertex immediately following u_i on C. If any two of these vertices are adjacent, say $a_i \sim a_j$, then we construct a longer cycle by using (a_i, a_j), the portions of C from a_i to u_j and a_j to u_i, and a u_i, u_j-path through H (see first illustration). This argument includes the case $a_i = u_{i+1}$ (see second illustration), so we also conclude that no a_i has a neighbour in H. Hence $\{a_1, \ldots, a_k\}$ plus a vertex of H forms an independent set of size $k + 1$, proving that $\alpha(G) > k$ as desired.



□

4.4 Tournaments

In the last section of this chapter, we consider the notion of Hamiltonicity in directed graphs, where each edge can only be traversed in a single direction.

Definition 4.21. A *tournament* is a directed graph obtained by assigning a direction to every edge of the complete graph. That is, it is an orientation of K_n.

In the context of tournaments, it only makes sense to consider directed paths and cycles, because every edge is present. So, a Hamilton cycle is a spanning directed cycle.

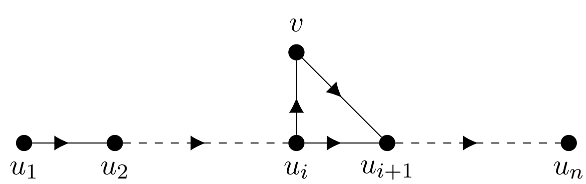
Theorem 4.22. *Every tournament has a Hamilton path.*

比大小? $v > u$ if (v,u)

$v_1 \cdots v_n$ $\exists\, u > v > r > u$ it doesn't matter

Proof. We proceed by induction on n. The case $n = 2$ is clear (there is only one tournament on two vertices, namely a single directed edge, which is already a Hamilton path).

Suppose the claim holds for all tournaments on n vertices, and let T have $n+1$ vertices. Let v be any vertex of T. Then $T \backslash \{v\}$ is also a tournament and hence has a Hamilton path $u_1 \to \cdots \to u_n$. Note that if $v \to u_1$, we are done as we have the path $v \to u_1 \to u_2 \to \ldots \to u_n$. So suppose $u_1 \to v$. Let i be the maximum index such that $u_i \to v$. If $i = n$ we are done, with the Hamilton path $u_1 \to \cdots \to u_n \to v$. If not then $v \to u_{i+1}$ and we can take the Hamilton path $u_1 \to \cdots \to u_i \to v \to u_{i+1} \to \cdots \to u_n$.



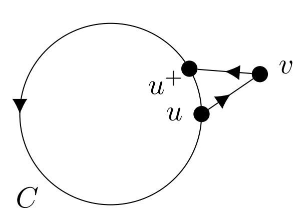
□

Definition 4.23. A tournament is *strongly connected* if for all $u, v \in V(T)$ there is a directed path from u to v.

Theorem 4.24. *A tournament T is strongly connected if and only if it has a Hamilton cycle.*

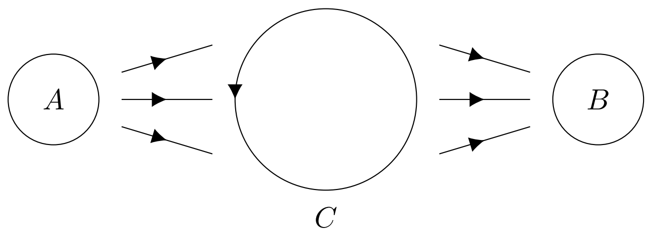
Proof. If T is Hamiltonian, then it is immediately clear that it is strongly connected: for any u and v simply consider the portion of a Hamilton path between u and v.

Now, suppose T is not Hamiltonian. Let C be a longest cycle in T and let $v \notin C$. If C has 2 consecutive vertices u, u^+ such that $u \to v$ and $v \to u^+$, then there is a longer cycle on the vertex set $C \cup \{v\}$:



Otherwise all edges between v and C go in only one direction, and we can assume this holds for all $v \notin C$. Let A be the set of $v \notin C$ such that edges go from v to C, and let B be the set of $v \notin C$ such that edges go from C to v. If one of A or B are empty, it immediately follows that T is not strongly connected: if say B is empty then there is no path from any vertex of C to any vertex of A.

So, suppose both of A and B are nonempty. There is no edge oriented from a vertex $b \in B$ to a vertex $a \in A$, because if there was we could extend C, contradicting maximality (we could replace any edge $x \to y$ in C with the path $x \to b \to a \to y$). But then there is no path from any vertex in B to any vertex in A, so again T is not strongly connected.



□

5 Matchings

Having discussed paths and cycles in the previous section, we now look at another important type of subgraph — the matching.

5.1 An introduction to matchings

Definition 5.1. A set of edges $M \subseteq E(G)$ in a graph G is called a *matching* if $e \cap e' = \varnothing$ for any pair of edges $e, e' \in M$ — that is, the edges are pairwise disjoint.

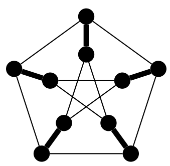
Example 5.2. Here is just a short list of situations where it is useful to think about and look for matchings.

- Suppose that a factory owns several machines, each of which can only be operated by qualified members of staff, and that each staff member can only run one machine at a time. This gives rise to a bipartite graph between workers and machines. If we want to have many machines operating at the same time, we need a large matching in this bipartite graph.
- Suppose we have a number of hour-long jobs to perform on two computers. Certain jobs can only be started once other jobs are finished. We can define a graph by putting an edge between every pair of jobs that can be performed simultaneously; to finish all the jobs as quickly as possible, we would like to find a large matching in our graph.
- The molecular structure of a compound can be described by a graph. For certain kinds of hydrocarbon molecules (*benzenoids*), a perfect matching of this graph gives information about the location of its "double bonds".
- When students apply to universities, each student has a list of university preferences, and each university also has a list of preferences for students. In order to decide which students should go to which university, we need to find a bipartite matching that is somehow compatible with these preferences. This kind of situation is called the *stable matching* problem, and is extremely important in economics and operations research. The Gale-Shapley algorithm for efficiently computing a stable matching earned its creators the 2012 Nobel prize in Economics.

- Algorithms to find large matchings are essential subroutines for solving other optimization problems as well. The *Chinese postman problem* involves visiting several designated points while travelling as short a total distance as possible. This problem can be efficiently solved by first solving a set of shortest path problems, then solving a certain matching problem.

Definition 5.3. We denote the maximum size of a matching in G by $\nu(G)$, the *matching number* of G. A matching is *perfect* if $|M| = \frac{1}{2}|V(G)|$, i.e., it covers all vertices of G.

Example 5.4.

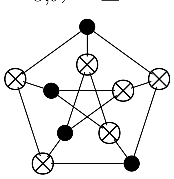
- $G = K_n$: $\nu(G) = \lfloor \frac{n}{2} \rfloor$
- $G = K_{s,t}$, $s \leq t$: $\nu(G) = s$
- $G =$  : $\nu(G) = 5$

Remark 5.5. A matching in a graph G corresponds to an independent set in the line graph $L(G)$.

A closely related notion to a matching (in quite a strong sense, as we shall see later) is that of a vertex cover.

Definition 5.6. A set of vertices $T \subseteq V(G)$ of a graph G is called a *cover* of G if every edge $e \in E(G)$ intersects T ($e \cap T \neq \varnothing$), i.e., $G \setminus T$ is an empty graph. The minimum size of a cover, called the *cover number*, is denoted by $\tau(G)$.

Example 5.7.

- $G = K_n$: $\tau(G) = n - 1$
- $G = K_{s,t}$, $s \leq t$: $\tau(G) = s$
- $G =$  : $\tau(G) = 6$.

  To see this, note that the graphs induced by the outer 5 vertices and inner 5 vertices are both 5-cycles C_5. Since $\tau(C_5) = 3$, at least 3 of the outer vertices and 3 of the inner vertices must be included in a vertex cover.

In our first general result, we bound the matching and cover numbers in terms of one another.

Proposition 5.8. $\nu(G) \leq \tau(G) \leq 2\nu(G)$.

Proof. Let M be a maximum matching in G. Since every cover has at least one vertex on each edge of M and edges are disjoint, we have $\nu(G) \leq \tau(G)$. Note also that since M is maximum, every edge $e \in E(G)$ intersects some edge $e' \in M$, otherwise we get a larger matching. So the vertices covered by M form a cover for G, hence $\tau(G) \leq 2|M| = 2\nu(G)$. $\square$

5.2 Hall's Theorem

In this section we introduce the following classic result, which gives a necessary and sufficient condition for the existence of maximum size matchings in bipartite graphs.

Theorem 5.9 (Hall 1935). *A bipartite graph $G = (V, E)$ with bipartition $V = A \cup B$ has a matching covering A if and only if*

$$|N(S)| \geq |S| \quad \forall S \subseteq A. \tag{1}$$

Proof. It is easy to see that if G has such a matching then (1) holds, as each vertex in $S \subseteq A$ must be matched to a distinct vertex in the neighbourhood $N(S)$.

To show the other direction, we apply (strong) induction on $|A|$. For $|A| = 1$ the assertion is true. Now let $|A| \geq 2$, and assume that (1) is sufficient for the existence of a matching covering A whenever $|A|$ is smaller.

First, let us suppose that the stronger inequality $|N(S)| \geq |S| + 1$ holds for every non-empty set $S \subsetneq A$. Then pick an edge $\{a, b\} \in G$ and consider the graph $G' = G \backslash \{a, b\}$ obtained by deleting the vertices a and b. Then every non-empty set $S \subseteq A \setminus \{a\}$ satisfies

$$|N_{G'}(S)| = |N_G(S) \setminus \{b\}| \geq |N_G(S)| - 1 \geq |S|,$$

so by the induction hypothesis G' contains a matching covering $A \setminus \{a\}$. Together with the edge $\{a, b\}$, this yields a matching covering A in G.

If this stronger inequality does not always hold, there must be a non-empty proper subset $A' \subsetneq A$ with neighbourhood $B' = N(A')$ such that $|A'| = |B'|$. By the induction hypothesis, $G' = G[A' \cup B']$ contains a matching covering A'. But $G \backslash G'$ satisfies (1) as well: for any set $S \subseteq A \setminus A'$ with $|N_{G \backslash G'}(S)| < |S|$, we would have $|N_G(S \cup A')| = |N_{G \backslash G'}(S)| + |B'| < |S \cup A'|$, contrary to our assumption. Again, by induction, $G \backslash G'$ contains a matching of $A \setminus A'$. Putting the two matchings together, we obtain a matching in G covering A. □

Corollary 5.10. *If a bipartite graph $G = (A \cup B, E)$ is k-regular with $k \geq 1$, then G has a perfect matching.*

Proof. If G is k-regular, then clearly $|A| = |B|$, since the total number of edges is $k|A| = \sum_{x \in A} d(x) = \sum_{y \in B} d(y) = k|B|$. It thus suffices to show by Theorem 5.9 that G contains a matching covering A. Now every set $S \subseteq A$ is joined to $N(S)$ by a total of $k|S|$ edges, and these are among the $k|N(S)|$ edges of G incident with $N(S)$. Therefore $k|S| \leq k|N(S)|$, so G does indeed satisfy (1). □

Corollary 5.11. *Every regular graph of positive even degree has a 2-factor (a spanning 2-regular subgraph).*

Proof. Let G be any connected $2k$-regular graph. By Theorem 4.5, G contains an Euler tour. Define a new graph G' by splitting every vertex v into two vertices v^- and v^+. If an edge of the Euler tour goes from v to w, put an edge in G' from v^+ to w^-. This gives a natural correspondence between the edges in G and those in G'. It is easy to see that G' is bipartite and k-regular, and so contains a perfect matching. Collapsing each pair of vertices v^-,v^+ back into a single vertex v, a perfect matching of G' corresponds to a 2-factor of G — each vertex v is incident to one edge which was incident to v^+ in G' and one edge incident to v^- in G'. □

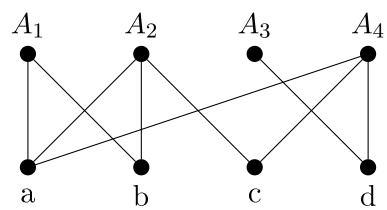
Remark 5.12. A 2-factor is a disjoint union of cycles covering all the vertices of a graph

Definition 5.13. Let $A_1, \ldots, A_n$ be a collection of sets. A family $\{a_1, \ldots, a_n\}$ is called a *system of distinct representatives* (SDR) if $a_i \in A_i$ for all i and all the a_i are distinct.

Corollary 5.14. *A collection $A_1, \ldots, A_n$ has an SDR if and only if for all $I \subseteq [n]$ we have $|\bigcup_{i\in I} A_i| \geq |I|$.*

Proof. Define a bipartite graph with parts $A = [n]$ and $X = \bigcup_i A_i$ such that $\{i, a\}$ is an edge if and only if $a \in A_i$. A matching of $[n]$ in this graph corresponds exactly to an SDR, where an edge $\{i, a\}$ in the matching means that $a_i = a$. But the condition $|\bigcup_{i\in I} A_i| \geq |I|$ for all $I \subseteq [n]$ is precisely Hall's condition for the existence of a matching covering A, so Hall's theorem provides the desired equivalence. □

Example 5.15. $A_1 = \{a, b\}$, $A_2 = \{a, b, c\}$, $A_3 = \{d\}$, $A_4 = \{a, c, d\}$.



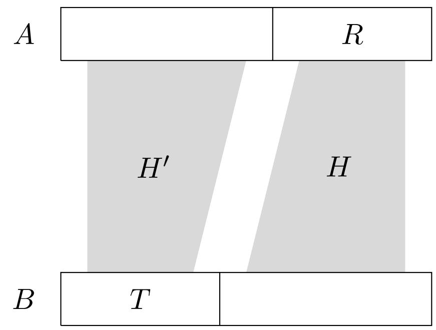
Our final result in this subsection establishes the equivalence between the matching and vertex cover numbers in bipartite graphs.

Theorem 5.16 (Kőnig 1931)**.** *If $G = (A \cup B, E)$ is a bipartite graph, then the maximum size of a matching in G equals the minimum size of a vertex cover of G.*

Proof. We have already seen that a minimum cover has at least the size of a maximum matching. Now take a minimum vertex cover U of G. We construct a matching of size $|U|$ to prove that equality can always be achieved.

Let $R = U \cap A$ and $T = U \cap B$. Let H, H' be the subgraphs of G induced by $R \cup (B\backslash T)$ and $T \cup (A\backslash R)$. We use Hall's theorem to show that H has a complete matching of R into $B\backslash T$ and H' has a complete matching of T into $A\backslash R$. Since these subgraphs are disjoint, the two matchings together form a matching of size $|U|$ in G.

Since $R \cup T$ is a vertex cover, G has no edge from $B\backslash T$ to $A\backslash R$. Suppose $S \subseteq R$, and consider $N_H(S) \subseteq B\backslash T$. If $|N_H(S)| < |S|$, then we can substitute $N_H(S)$ for S in U and obtain a smaller vertex cover, since $N_H(S)$ covers all edges incident to S that are not covered by T. The minimality of U thus implies that Hall's condition holds in H, and hence H has a complete matching of R into $B\backslash T$. Applying the same argument to H' yields the rest of the matching.



□

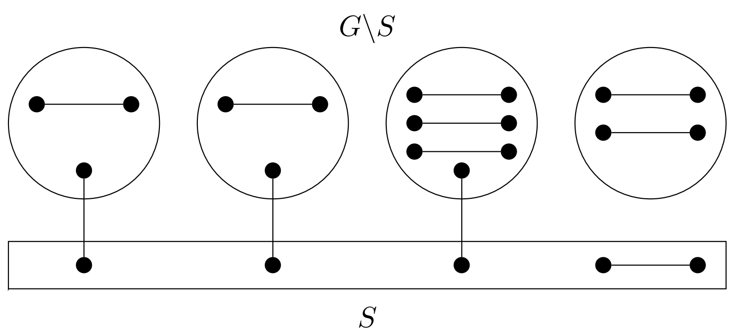
5.3 Tutte's Theorem

Hall's Theorem characterises the bipartite graphs with perfect matchings. However, sadly,[1] not all graphs are bipartite. Tutte's Theorem tells us when an arbitrary graph has a perfect matching.

Given a graph G, let $q(G)$ denote the number of its *odd components*, i.e. the ones of odd order. If G has a perfect matching then clearly

$$q(G\backslash S) \leq |S| \quad \text{for all } S \subseteq V(G), \tag{2}$$

since every odd component of $G\backslash S$ will send an edge of the matching to S, and each such edge covers a different vertex in S.



Theorem 5.17 (Tutte, 1947)**.** *A graph G has a perfect matching if and only if $q(G\backslash S) \leq |S|$ for all $S \subseteq V(G)$.*

Proof. As noted above, Tutte's condition is necessary; we prove sufficiency. Tutte's condition is preserved by addition of edges: if $G' = G \cup \{e\}$ and $S \subseteq V(G)$, then $q(G'\backslash S) \leq q(G\backslash S)$, because when the addition of e combines two components of $G\backslash S$ into one, the number of components that have odd order does not increase. Therefore, it suffices to consider a simple graph G such that G satisfies (2), has no perfect matching, but adding any edge to G creates a perfect matching. We will obtain a contradiction in every case by constructing a perfect matching in G.

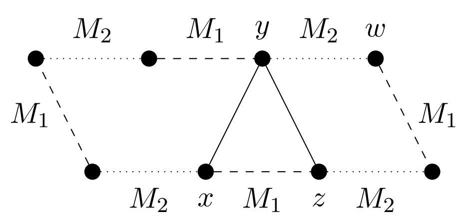
By considering $S = \varnothing$, we know that G has an even number of vertices, since a graph of odd order must have a component of odd order. Let U be the set of vertices in G that are connected to all

[1] Or perhaps happily, because otherwise life would be easy and boring.

other vertices. Suppose $G\backslash U$ consists of disjoint complete graphs; we build a perfect matching for such a G. The vertices in each component of $G\backslash U$ can be paired up arbitrarily, with one left over in the odd components. Since $q(G\backslash U) \leq |U|$ and each vertex of U is adjacent to all of $G\backslash U$, we can match these leftover vertices arbitrarily to vertices in U to complete a matching.

This leaves the case where $G\backslash U$ is not a disjoint union of cliques. We can therefore find two vertices in the same component which are not adjacent, and on a shortest path between them there are two nonadjacent vertices x, z at distance 2 (which have a common neighbour y). Furthermore, $G\backslash U$ has another vertex w not adjacent to y, since $y \notin U$. By the maximality of G, adding any edge to G produces a perfect matching. Let M_1 and M_2 be perfect matchings in $G \cup \{x, z\}$ and $G \cup \{y, w\}$, respectively. It suffices to show that in $M_1 \cup M_2$ we can find a perfect matching avoiding $\{x, z\}$ and $\{y, w\}$, because that would be contained in G.

Let F be the graph on $V(G)$ with the edges that belong to exactly one of M_1 and M_2. Note that F contains $\{x, z\}$ and $\{y, w\}$. Since every vertex of G has degree 1 in each of M_1 and M_2, every vertex of G has degree 0 or 2 in F. Hence F is a collection of disjoint even cycles (alternating between edges of M_1 and M_2) and isolated vertices. Let C be the cycle of F containing $\{x, z\}$. If C does not also contain $\{y, w\}$, then the desired matching consists of the edges of M_2 from C and all of M_1 not in C. If C contains both $\{y, w\}$ and $\{x, z\}$, as illustrated below, then we use the edge $\{y, x\}$ or the edge $\{y, z\}$ to obtain a matching of $V(C)$ using only edges of G (avoiding both $\{x, z\}$ and $\{y, w\}$). Specifically, we use $\{y, x\}$ if the distance between y and x in C is odd, and we use $\{y, z\}$ otherwise (then the distance between y and z in C is odd). In the illustration below, this second case applies. The remaining vertices of C form two paths of even order. We use the edges of M_1 in one of these paths and the edges of M_2 in the other to produce a matching in C that does not use $\{x, z\}$ or $\{y, w\}$. (In the illustration below, we use the edges of M_1 on the right side of $\{y, z\}$ and the edges of M_2 on the left). Combined with M_1 or M_2 outside C, we have a perfect matching of G.



□

Corollary 5.18 (Petersen, 1891). *Every 3-regular graph with no cut-edge has a perfect matching.*

Proof. Let $S \subseteq V(G)$. Let H be a component of $G\backslash S$, with $|H|$ odd. The number of edges between S and H cannot be 1, since G has no cut-edge. It also cannot be even, because then the sum of the vertex degrees in H would be odd. Hence there are at least three edges from H to S.

Since G is 3-regular, each vertex of S is incident to at most three edges between S and $G\backslash S$. Combining this fact with the previous paragraph, we have $3q(G\backslash S) \leq 3|S|$ and hence $q(G\backslash S) \leq |S|$, which proves the corollary. □

Example 5.19. The condition that the graph has no cut-edge is necessary. The graph below is 3-regular but has no perfect matching. Deleting the central vertex leaves 3 odd components.