

Course Notes

Graph Theory

Alex Rutar

BSM Fall 2018

Contents

1	Basic Structure of Graphs	3
1.1	A Brief Introduction	3
1.1.1	Basic Definitions	3
1.2	Paths, Circuits, and Cycles	4
1.2.1	Eulerian Circuits	4
1.2.2	Hamiltonian Cycles	5
1.2.3	Necessary Conditions	5
1.2.4	Sufficient Conditions	6
1.3	Matchings	8
1.3.1	Bipartite Matchings	9
1.3.2	Minimax: Matchings and Coverings	10
1.3.3	Tutte's Theorem and Applications	11
1.3.4	Stable Matchings	12
1.4	Colourings	13

Chapter 1

Basic Structure of Graphs

1.1 A Brief Introduction

1.1.1 Basic Definitions

Def'n. 1.1.1 A **graph** $G = (V, E)$ consists of a vertex set V and edge set E where $E \subseteq \binom{V}{2}$.

Note that we write $\binom{V}{2}$ instead of $V \times V$ to make it clear that we cannot have loops and multiple edges.

Def'n. 1.1.2 Two graphs F and G are **isomorphic** if there exists a bijective mapping $f : V(F) \rightarrow V(G)$ such that for every $a, b \in V(F)$: $\{a, b\} \in E(F) \Leftrightarrow \{f(a), f(b)\} \in E(G)$.

Def'n. 1.1.3 The **degree** of a vertex $v \in V(G)$ is the number of edges having v as an endpoint. We denote this number by $d(v)$, and $\Delta(G)$ to denote the maximal degree in G , and $\delta(G)$ to denote the minimal degree in G .

Def'n. 1.1.4 The **neighbourhood** of a graph U is denoted by $N(U) := \{v \in V(G) : \exists u \in U \text{ s.t. } \{v, u\} \in E(G)\}$.

Def'n. 1.1.5 A **path** is a sequence $v_1 e_1 v_2 e_2 \dots v_i e_i v_{i+1} \dots e_j v_{j+1}$ where each $v_i \in V(G)$ and $e_i = \{v_i, v_{i+1}\} \in E(G)$ where all v_i 's are different. A **cycle** is a path in which $v_1 = v_{j+1}$.

Def'n. 1.1.6 The **complementary graph** of $G = (V, E)$ is $\overline{G} = (V, \binom{V}{2} \setminus E)$.

Def'n. 1.1.7 A graph G is called **connected** if for every $u, v \in V(G)$ there exists a path between u and v .

Def'n. 1.1.8 A connected graph that becomes disconnected with the removal of any edge is called a **tree**. Equivalently, a **tree** is a graph which is connected and contains no cycle.

Prop. 1.1.9 Any tree on at least two vertices contains at least two vertices of degree 1 ("leaves").

PROOF Consider a path of maximal length. We claim that both endpoints of P have degree one. Suppose for contradiction that an endpoint has greater than one. Then the endpoint has another neighbour on the path (in which case we have a cycle), or a unique neighbour (in which case the path is not maximal), a contradiction in either case. \square

Prop. 1.1.10 *A tree on n vertices always has $n - 1$ edges.*

PROOF Delete the edges one by one. Each time, the number of connected components increases by one. After deleting all edges, we have n components, at the beginning, we have one, so we deleted $n - 1$ edges. \square

We now have an interesting question: how many different trees can be given on n labelled vertices? To investigate this, we consider the Prüfer code. Delete the smallest labelled degree one vertex and write up its unique neighbour's label. Continue doing this until only one point remains. The obtained sequences of labels is the Prüfer code.

Properties:

- The length is $n - 1$
- The last digit must be n

Thm. 1.1.11 (Cayley) *The number of different trees on n labelled vertices is n^{n-2} .*

PROOF We will show that sequences $x \in \{1, 2, \dots, n\}^{n-1}$ with $x_{n-1} = n$ are in bijective correspondence with the trees on n labelled vertices. First, given $a_1 a_2 \dots a_{n-2} a_{n-1}$ with $a_{n-1} = n$ we want to decode it. Let b_1, b_2, \dots, b_{n-1} be the sequence of labels of vertices deleted in the order of the indices. If we “decode” $b_1 b_2 \dots b_{n-1}$, we know the tree, since we have the $n - 1$ edges $\{a_i, b_i\}$.

$$1. \ b_1 := \min\{k \in \{1, 2, \dots, n\} : k \notin \{a_1, \dots, a_{n-1}\}\}$$

$$2. \ b_2 := \min\{k \in \{1, 2, \dots, n\} : k \notin \{b_1, a_2, \dots, a_{n-1}\}\}$$

$$(*) \quad b_i := \min\{k \in \{1, 2, \dots, n\} : k \notin \{b_1, \dots, b_{i-1}, a_i, \dots, a_{n-1}\}\}$$

We show that taking any sequence $a_1 a_2 \dots a_{n-1}$ with $a_{n-1} = n$ and applying $(*)$ to obtain b_1, \dots, b_{n-1} , the graph we obtain on vertices $1, \dots, n$ with the $n - 1$ edges $\{a_i, b_i\}$ (1) is a tree, and (2) has Prüfer code is just $a_1 a_2 \dots a_{n-1}$.

Note that $(*)$ implies that $\{b_1, b_2, \dots, b_{n-1}, a_{n-1}\} = \{1, 2, \dots, n\}$. Define graphs T_i for $i = n - 1, n - 2, \dots, 2, 1$ on the graph spanned by the edges $\{a_{n-1}, b_{n-1}\}, \{a_{n-2}, b_{n-2}\}, \dots, \{a_i, b_i\}$. It suffices to prove that T_i is a tree for every i and b_i is its smallest labelled degree 1 vertex.

We do this by induction. Clearly, it is true for $i = n - 1$. Once it is true for $i = n - 1, \dots, j + 1$, we prove this for $i = j$. We know that T_{j+1} is a tree, and we wish to add the edge $\{b_j, a_j\}$. Thus $b_j \notin V(T_{j+1}) = \{b_{j+1}, b_{j+2}, \dots, b_{n-1}, a_{n-1}\}$ so b_j is indeed degree one; $a_j \in V(T_{j+1})$ and T_j is a tree. If b_j was not the smallest degree 1 vertex, then there exists some $k > j$ such that $b_k < b_j$ and b_k has degree one in T_j . But then $b_k \notin \{b_1, \dots, b_{j-1}, a_j, \dots, a_{n-1}\}$ so $(*)$ would have chosen it in place of b_j , a contradiction. \square

1.2 Paths, Circuits, and Cycles

1.2.1 Eulerian Circuits

Königsberg (modern Kaliningrad)

Def'n. 1.2.1 *An **Eulerian circuit** is a closed walk in a graph that contains every edge exactly once. An **Eulerian path** is a walk containing every edge exactly once and not (necessarily) ending at the same vertex.*

Note that we do allow multiple edges between vertices (graph is necessary simple). We also assume that our graph is connected.

Thm. 1.2.2 (Euler) *A graph contains an Eulerian circuit if and only if every vertex has even degree.*

Cor. 1.2.3 *A graph contains an Eulerian path if and only if all but two vertices have even degree.*

In both cases, necessity is obvious: every time a path arrives at a vertex, it adds two to the degree since there is a unique edge in and out from the vertex. Thus for the corollary the only odd vertices can be the endpoints, and for the theorem, there are no endpoints in the path.

PROOF (COR.) To see the corollary, first add an edge connecting the two odd degree vertices. Then by the theorem, we have an Eulerian circuit walk through it so that the added edge is the last one traversed. Delete it, and the remaining part of the walk gives our Eulerian path \square

PROOF (THM.) We can now prove the theorem. Consider a maximal walk P on G that does not repeat edges. Because of the evenness of all degrees, it must be closed. If every edge is contained, it is an Eulerian circuit and we are done. If there exists some $e \in E(G)$ that is not in the walk, then by connectedness, there must be a path from a vertex in the walk to an endpoint of e . Considering a shortest such path (perhaps containing 0 edges), all edges on it are unused so far. Now take the starting point of this path on our walk, go through the closed path on our walk, go through the closed walk we have starting here and thus continue on the said path and include e , a contradiction. \square

1.2.2 Hamiltonian Cycles

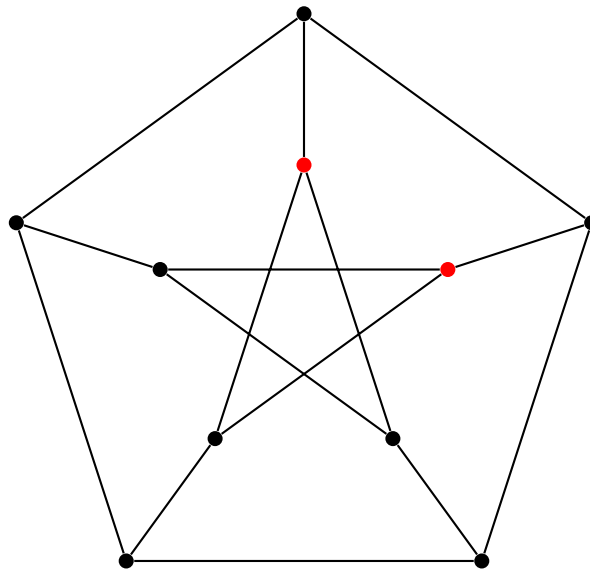
Def'n. 1.2.4 *A **Hamiltonian cycle** in a graph is a cycle containing every vertex exactly once. A **Hamiltonian path** is a path that contains every vertex exactly once.*

1.2.3 Necessary Conditions

Prop. 1.2.5 *If G contains a Hamiltonian cycle, then after deleting any k of its vertices, the remaining graph cannot have more than k components. Similarly, if G contains a Hamiltonian path, then deleting any k of its vertices yields a graph with at most $k + 1$ components.*

PROOF This can be easily proven by induction. \square

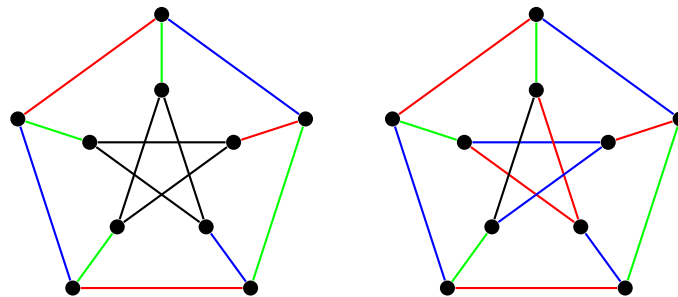
Here is a graph for which this condition holds, but does not have a Hamiltonian cycle.



The condition holds: delete $k = l_1 + l_2$ vertices, where l_1 is in the “outer cycle” and l_2 is in the “inner cycle”. (Above, $l_2 = 2$ and $l_1 = 0$.) Then the outer cycle may not fall apart into more than l_1 components, the inner cycle may not fall apart into more than l_2 component, so if $l_1, l_2 > 0$. If l_1 or l_2 is 0, then the graph remains connected.

Furthermore, there does not exist a Hamiltonian cycle. Suppose such a cycle exists, then it is a cycle containing 10 edges. Colour the alternating edges red and blue. Thus every vertex is adjacent to a red edge and a blue edge. Colour the remaining 5 edges white, so every vertex will be the end vertex of exactly 1 white edge. However, such an edge-coloring is impossible.

Up to isomorphism, the outer edges must be coloured with two of colour 1, two of colour 2, and one of colour 3. We then fill in the interior edges as required, until the inner cycle, in which we are forced to draw the following edges, yielding our contradiction.



1.2.4 Sufficient Conditions

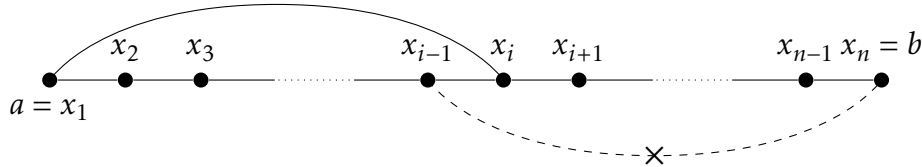
Thm. 1.2.6 (Dirac, 1952) *If G is a graph on n vertices, with every degree being at least $\frac{n}{2}$, then G contains a Hamiltonian cycle.*

As a fun fact, this is not Paul Dirac, but rather Gabriel Andrew Dirac (Paul Dirac’s stepson). Note that we must have $f(n) \geq n/2$. If not, then the graph composed of two disconnected components $K_{n/2}$ has degree $n/2 - 1$ everywhere but does not have a hamiltonian cycle. We also have the following stronger theorem:

Thm. 1.2.7 (Ore, 1960) *If G is a graph on n vertices such that for every nonadjacent pair of vertices $u, v \in V(G)$, $d(u) + d(v) \geq n$ is satisfied, then H contains a Hamiltonian cycle.*

PROOF Assume for contradiction we have G_0 satisfying the condition but does not have a Hamiltonian cycle. Saturate G_0 to obtain G : if two vertices are non-adjacent and connecting them still creates a Hamiltonian cycle, then connect them. Observe that the new graph is still a counterexample since the condition will obviously remain true. Do this maximally, so that the addition of any edge would create a Hamiltonian cycle.

Now, if $a, b \in V(G)$ are non-adjacent in G , then there exists a Hamiltonian path starting at a and ending at b . Consider such a Hamiltonian path (x_1, x_2, \dots, x_n) .



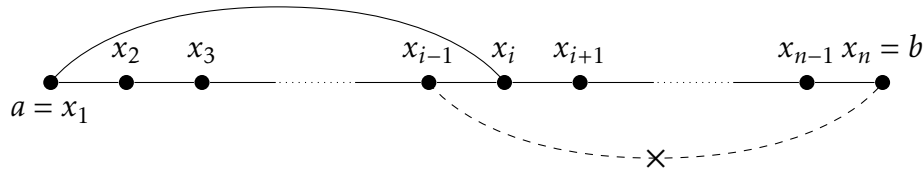
Observe that if $\{a, x_i\} \in E(G)$, then $\{a, x_{i-1}\} \notin E(G)$ or we would have a Hamiltonian cycle given by $(a, x_i, x_{i+1}, \dots, b, x_{i-1}, x_{i-2}, \dots, a)$. This implies that $d(b) \leq n - 1 - d(a)$: for each neighbour of a , there is a distinct non-neighbour of b . But this is a contradiction since $d(a) + d(b) \leq n - 1$ while $\{a, b\} \notin E(G)$ and G does not satisfy the condition. \square

Thm. 1.2.8 (Pósa, 1962) Let G be a graph on n vertices with degrees $d_1 \leq d_2 \leq \dots \leq d_n$. Then for every $k < n/2$, we have $d_k \geq k + 1$, then G contains a Hamiltonian cycle.

Thm. 1.2.9 (Chvátal, 1972)

- (i) Let G be a graph on n vertices with degrees $d_1 \leq d_2 \leq \dots \leq d_n$. Assume that whenever for some $k < n/2$, if we have $d_k \leq k$, $d_{n-k} \geq n - k$. Then G contains a Hamiltonian cycle.
- (ii) Assume that the number $d'_1 \leq d'_2 \leq \dots \leq d'_n$ do not satisfy the implication. Then there exists a graph G with degrees $d_1 \leq d_2 \leq \dots \leq d_n$ such that $d_i \geq d'_i$ and G has no Hamiltonian cycle.

PROOF (i) Assume that the statement is false, consider a counterexample G_0 , and saturate it to obtain G . This works: if an edge can be added without creating a Hamiltonian cycle; then after adding we will still have a counterexample, the validity of the condition cannot be destroyed by adding an edge. From Ore's proof, we know that for every $a, b \in V(G)$ that are non-adjacent, there exists a Hamiltonian path from a to b and $d(a) + d(b) \leq n - 1$. Consider an $a, b \in V(G)$, $\{a, b\} \notin E(G)$ pair for which $d(a) + d(b)$ is maximal



We may assume $d(a) \leq d(b)$. Then we have $d(a) \leq \frac{n-1}{2} < \frac{n}{2}$. For notation, $h = d(a)$. We claim that $d_h \leq h < \frac{n}{2}$. We show $d_h \leq h$.

From the Ore proof argument, we know that b has at least as many non-neighbours as many neighbours a has, i.e. h . Each of these non-neighbours x of b could have been chosen for a (as a non-adjacent pair for b), but it was not, so $d(x) \leq d(a) = h$. Thus there

exists an index greater than or equal to h with degree less than or equal to h . This exactly means $d_h \leq h$, and we have our claim.

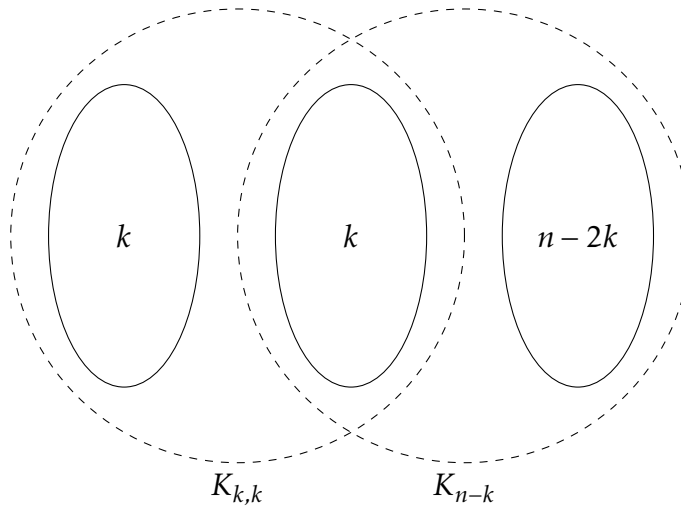
Thus by our condition, $d_{n-h} \geq n-h$, so there exist $h+1$ degrees $\geq n-h$. Then by the Pidgeonhole Principle, there exists at least one non-neighbour of a , call it y , that has degree $\geq n-h$. But then a, y is a non-adjacent pair with $d(a) + d(y) \geq h + n - h = n > d(a) + d(b)$, a contradiction to the choice of a, b .

- (ii) Assume $d'_1 \leq \dots \leq d'_n$ violates our condition. Then we have k so that $d'_k \leq k < n/2$ and $d'_{n-k} \leq n-k-1$. Fix such a k , then

$$d'_1 \leq d'_2 \leq \dots \leq d'_k \leq k$$

$$d'_{k+1} \leq d'_{k+2} \leq \dots \leq d'_n \leq n-k-1$$

and clearly, $d'_{n-k+1} \leq \dots \leq d'_n \leq n-1$. Set $d_1 = d_2 = \dots = d_k := k$, $d_{k+1}, d_{k+2} = \dots = d_{n-k} := n-k-1$ and $d_{n-k+1} = \dots = d_n := n-1$. Thus it is enough to show a graph with these degrees and no Hamiltonian cycle. Here is such a graph:



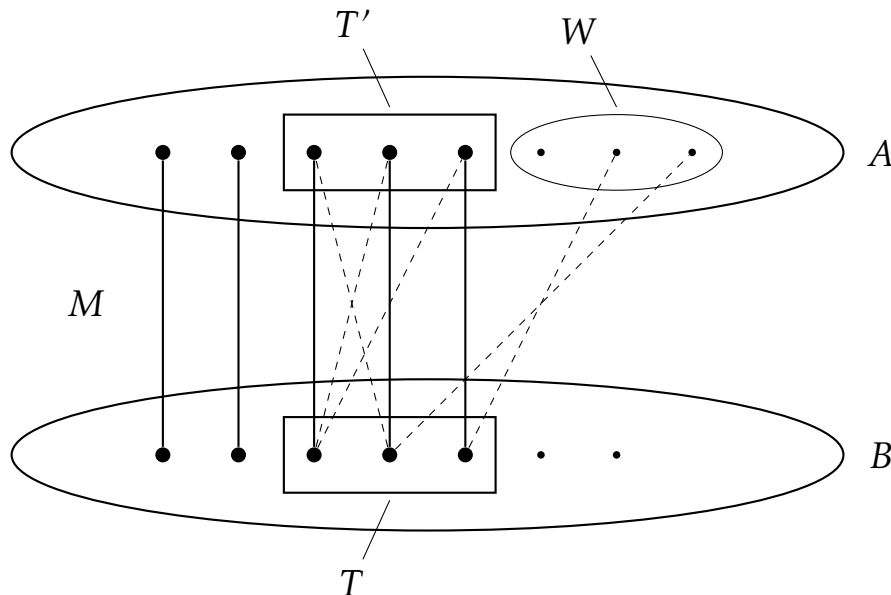
It can be verified that the degrees in the first component are k , second are $n-1$ and third are $n-k-1$. Furthermore, the graph has no Hamiltonian cycle. Remove the k vertices of degree $N-1$, and we are left with $k+1$ components, so the necessary condition for Hamiltonicity we had is violated. \square

1.3 Matchings

Def'n. 1.3.1 A **matching** in a graph is a subset $M \leq E(G)$ such that all edges in $E(G)$ are independent, meaning that if $e, f \in M$, then $e \cap f = \emptyset$.

Def'n. 1.3.2 A matching is **maximal** if no edges can be added to it, **maximum** if there is no larger matching, and **perfect** if it covers all the vertices.

PROOF (HALL) Let $M \subseteq E(G)$ be a matching in a bipartite graph $G = (A, B; E)$ for which there exists no augmenting path. Assume M does not cover A , let the set of vertices in A not covered by M be called W , so $W \neq \emptyset$. Let $T \subseteq B$ be the set of vertices one can go to from W via an alternating path (w.r.t. M).



Every $v \in T$ is covered by M , otherwise there would exist an augmenting path w.r.t. M , but we assumed it does not exist. Let T' be the set of vertices in A that are the pairs of the vertices in T according to M . We have $|T'| = |T|$, and we claim that $U = T' \cup W$ violates the condition. In particular, we show that $N(T' \cup W) \subseteq T$. Indeed, $N(T' \cup W) = N(T') \cup N(W)$.

1. $N(W) \subseteq T$ since any single edge starting at some $w \in W$ can ... as an alternating path, so its other end must be in T by definition (of T).
2. $N(T') \subseteq T$. Let $v \in T'$; then v has a matched pair $v' \in T$ (i.e., $\{v, v'\} \in M$) and since $v' \in T$, there exists an alternating path from W to v' . Thus path does not contain the matching edge $\{v, v'\}$, so $P \cup \{v, v'\}$ is still an alternating path and if z is a neighbour of v' not yet in the path, then $P \cup \{v, v'\} \cup \{v', z\}$ is as an alternating path that goes from W to z . Since $z \in T$ by definition, we have $N(T') \subseteq T$.

Thus $N(T' \cup W) \subseteq T$. But $|T| = |T'| < |T'| + |W| = |T' \cup W|$, violating Hall's condition. Thus if no subset of A violates the condition, then there still exists an augmenting path, so one can increase the size of M . If we cannot do this anymore, then either the condition is violated, or $W = \emptyset$ and M contains every vertex in A . \square

1.3.2 Minimax: Matchings and Coverings

Def'n. 1.3.9 The *matching number* $\nu(G)$ of a graph G is the size of the number of edges in a maximum matching.

Def'n. 1.3.10 The *(vertex) covering number* $\tau(G)$ of a graph G is the minimum number of vertices that "cover" all edges, i.e. has the property that every edge is represented by at least one of them.

Note that $\nu(G) \leq \tau(G)$ since every edge in a matching needs a separate point in a vertex covering. Strict inequality can occur; for example any odd cycle. This is an example of a “minimax” property: if equality holds, then both values must be optimal.

Thm. 1.3.11 (König) *If G is bipartite, then $\tau(G) = \nu(G)$.*

PROOF Let M be a maximum matching in a bipartite graph G , i.e. $|M| = \nu(G)$. Define W, T, T' as in the proof of Hall’s Theorem. We proved that $N(T' \cup W) = T$. This implies that the vertices in T cover all edges with an endpoint in $T' \cup W$. Since every edge has an endpoint in A , the remaining edges are covered by the vertices in $A \setminus (T' \cup W)$. Thus $\tau(G) \leq |T| + |A \setminus (T' \cup W)| = |M|$ and equality holds. \square

1.3.3 Tutte’s Theorem and Applications

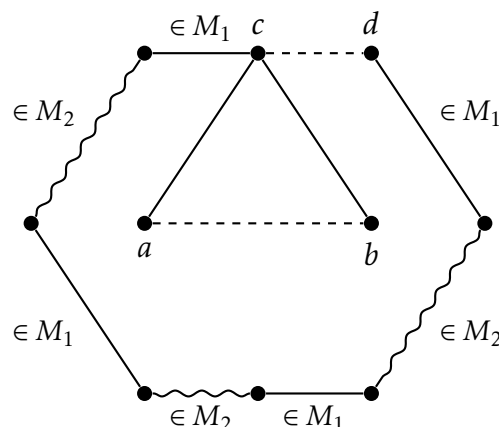
Thm. 1.3.12 (Tutte) *A graph G contains a perfect matching iff for any $S \subseteq V(G)$, we must have $c_{\text{odd}}(G \setminus S) \leq |S|$. c_{odd} denotes the number of components with odd size.*

PROOF (LOVÁSZ) (\Rightarrow) In the odd components, (i.e. components with an odd number of vertices) of $G \setminus S$ at least one vertex must be matched to a vertex in S . Thus $|S|$ cannot be smaller than the number of such components, which is $c_{\text{odd}}(G \setminus S)$.

(\Leftarrow) Assume the statement is false and let G_0 be a counterexample. Saturate G_0 : add edges until impossible without creating a perfect matching. We must show that we cannot increase the number of odd components by adding edges. If the edge joins two even components, then we get a new even component; or two odd components, and we have a new even component; or an even and an odd component, and we have a new odd component. In any case, the number of odd components stays the same and the graph remains a counterexample. Let the saturated graph be denoted by G .

It is sufficient to prove that deleting the points of G that are connected to all other vertex, the remaining graph is the union of vertex disjoint graphs. Suppose this is true: then in $G \setminus S$ (where $S := \{v \in V(G) : \{u, v\} \in E(G) \forall u \in V(G), u \neq v\}$), we can create a perfect matching within each even component and every vertex except one in the odd components (since the components are complete). We can then pair these components arbitrarily in S , since S is connected to everything. Then there must be only an even number of vertices in S remaining, or G violates the condition with $S = \emptyset$; and these vertices are mutually connected.

Thus assume this claim is not true. Then there exist vertices $a, b, c \in V(G)$ inducing a path on 3 vertices. Also, $c \notin S$. Thus we have some $d \in V(G)$ so that $\{c, d\} \notin E(G)$.



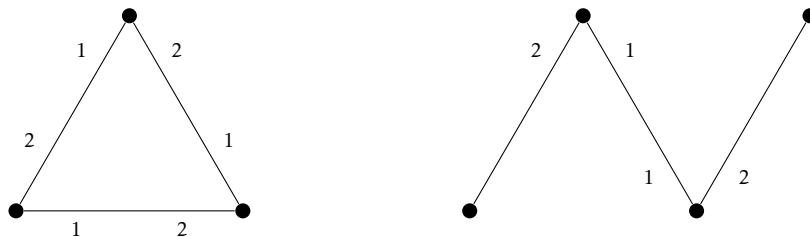
Since G is saturated, there exists a perfect matching M_1 in $G \cup \{a, b\}$ and M_2 in $G \cup \{c, d\}$. Start a walk from d along alternating edges in M_1 and M_2 . Let P be a longest such path. Then P can end only in a , b , or c .

First consider that we end in c . Then the edge set $M_2 \setminus P \cup (M_1 \cap P)$ forms a perfect matching. If P ends in a or b , then without loss of generality we may assume it ends in a . Now consider the cycle $C = P \cup \{a, c\} \cup \{c, d\}$. Then $M_2 \setminus C \cup (M_1 \cap C)$ is a perfect matching. Thus a perfect matching exists in G , so the claim is true, so the theorem holds. \square

1.3.4 Stable Matchings

Def'n. 1.3.13 Let G be a graph with a linear order of the neighbours of $v \in V(G)$ for every $v \in V(G)$ (these are called **preference lists**). A matching $M \subseteq E(G)$ is said to be **stable** with respect to these preference lists if there exists no edge in $e \in E(G) \setminus M$, $e = \{a, b\}$, such that both a and b prefer each other more than their current pair according to M .

A stable matching may not exist: for example, cyclic preferences in K_3 . As well, a stable matching may not be a maximum matching (though it must be maximal).



Thm. 1.3.14 (Gale-Shapley) If G is bipartite, then there always exists (i.e. for any preference lists) a stable matching.

PROOF We give an algorithm that produces a stable matching. It works in rounds.

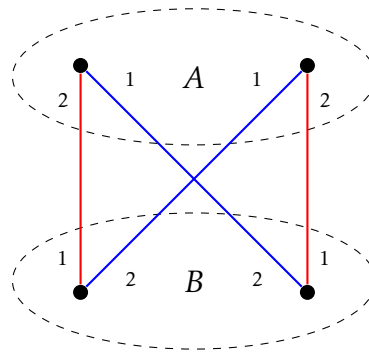
During the first round, each $b \in B$ “offers” a pairing to the neighbour that is first on their preference list. Then each $w \in W$ chooses the neighbour that is highest on their preference list, offers a wait to that neighbour, and denies all other offers.

During the second round, each $b \in B$ who got an answer of “wait” will do nothing, and those who got denied make an offer to the second on their preference list. Now for each $w \in W$, do as in round 1: say “wait” for the best, and “no” to the others.

Repeat this process until there are no more new offers made. Clearly, this is a matching, so it suffices to show it is stable. It is also stable: suppose there is an $e \in E(G) \setminus M$ where M is the matching obtained. There can be two reasons $e = \{b, w\} \notin M$:

1. b never offered to w , but then it is because b got accepted by someone he prefers. Thus e is not a “destabilizing” edge.
2. b offered to w but got rejected. But then this happens because w got a better offer (according to her preference list), so she has a better partner. Thus e is not a “destabilizing” edge. \square

Interestingly, this leads to an optimal matching for the set B , but not for the set W . For example:



Both the red and blue matchings are stable, but the red matching favours B , while the blue matching favours A .

1.4 Colourings

Def'n. 1.4.1 A **proper coloring** of a graph is a function $c : V(G) \rightarrow \mathbb{N}$ such that, for any $\{u, v\} \in E(G)$, $c(u) \neq c(v)$.

Def'n. 1.4.2 The minimum number of colours needed for a proper colouring of G is called the **chromatic number** $\chi(G)$ of G .

Def'n. 1.4.3 The **clique number** of G is denoted $\omega(G) := \max\{r : K_r \subseteq G\}$, in other words the size of the maximal complete subgraph of G .

Rmk. 1.4.4 A graph is bipartite if and only if $\chi(G) = 2$.

We certainly have $\chi(G) \leq \Delta(G) + 1$ by choosing colours greedily. However, greedy colourings certainly do not yield optimal colourings. For example, consider $K_{n,n} - nK_2$. We can refine this argument so that $\chi(G) \leq \max\{\delta(F) : F \subseteq G\} + 1$. We also have $\chi(G) \geq \omega(G)$.