NDMI011 Combinatorics and Graph Theory

Notes by

Marcel K. Goh (Prague, Czech Rep.)

10 June 2019

1. PRELIMINARIES

We say that a function $f(n)$ is $O\big(g(n)\big)$ (read "big-oh of $g(n)$") if there exist constants n_0 and C such that for all $n \geq n_0$, $f(n) \leq C \cdot g(n)$. If the ratio $\frac{f(n)}{g(n)}$ approaches 0 as n approaches infinity, then we say $f(n)$ is $o\big(g(n)\big)$ (read "little-oh of $g(n)$"). If $f(n)$ is both $O\big(g(n)\big)$ and $o\big(g(n)\big)$, then $f(n)$ is $\Theta\big(g(n)\big)$ (read "big-theta of $g\big(n)\big)$"). Finally, if the ratio $\frac{f(n)}{g(n)}$ approaches 1 as n approaches infinity, then we write $f(n) \sim g(n)$.

Let $G = (V, E)$ be a graph. The graph $G' = (V', E')$ is a *subgraph* of G if $V' \subseteq V$ and $E' \subseteq E$. A it cycle is a graph on $v_1, \ldots, v_n$ with edges $\{v_1, v_2\}, \ldots, \{v_{n-1}, v_n\}$. A *tree* is a connected graph that contains no cycle (i.e does not have a cycle as a subgraph).

Theorem. *In any graph $G = (V, E)$,*

$$\sum_{v \in V} \deg v = 2 \cdot |E|$$

Proof. Every edge $\{u, v\} \in E$ adds 1 to the degree of u and adds 1 to the degree of v. Hence if you sum over all vertices, each edge gets counted twice. ∎

Theorem. *The number of vertices with odd degree is even.*

Proof. Divide V into V_1 and V_2 where V_1 is the set of vertices with odd degree and V_2 is the set of vertices with even degree. Then we have $\sum_{v \in V} \deg v = \sum_{v \in V_1} \deg v + \sum_{v \in V_2} \deg v$. Observe that the sum over all odd vertices $\sum_{v \in V_1} \deg v$ equals $2 \cdot |E| - \sum_{v \in V_2} \deg v = 2(|E| - x)$ for some x, meaning it is even. Since the sum is even but each vertex has an odd degree, the number of vertices must be even. ∎

Theorem. *A tree with n vertices has $n - 1$ edges.*

Proof. Let $G = (V, E)$ be a tree and let n denote $|V|$, the number of vertices. Using induction on n, we show that it has $n - 1$ vertices. The base case $n = 1$ is easy, since a tree with only one vertex has no edges. Now assume that a tree with n vertices has $n - 1$ edges and we consider the case where G has $n + 1$ vertices. Let us create G' by removing a vertex from G. We cannot remove an internal vertex, since then G' would not be connected. So we must remove a leaf, along with the edge that connected it to G. Now G' has n vertices, so by the induction hypothesis it has $n - 1$ edges. This implies that G had n edges. ∎

2. GENERATING FUNCTIONS

2.1. Power series and generating functions

Given an infinite sequence $a_0, a_1, \ldots = (a_i)_{i=0}^{\infty}$, the power series

$$\sum_{i=0}^{\infty} a_i x^i$$

is called the *generating function* of $(a_i)_{i=0}^{\infty}$ and is denoted $a(x)$. For example, the sequence $1, 1, 1, \ldots$ has generating function $\sum_{i=0}^{\infty} x^i$, which converges to $1/(1 - x)$ for $x \in (-1, 1)$. Since we only really care about the coefficients of a generating function, we assume that x is within the power series' interval of convergence.

2.2. Binomial coefficients

For $r \in \mathbf{R}$ and $k \in \mathbf{N}$, the *binomial coefficient* is given by

$$\binom{r}{k} = \left(\prod_{i=0}^{k-1} (r - i) \right) / k!$$

where $\binom{r}{0} = 1$. If $r \in \mathbf{N}$, then we can think of $\binom{r}{k}$ as the number of ways to choose k objects out of a pool of r objects.

Theorem (*Generalised binomial theorem*). *For any $r \in \mathbf{R}$,*

$$\sum_{i \geq 0} \binom{r}{i} x^i = (1+x)^r.$$

Proof. In this proof we will gloss over technicalities relating to convergence. Let $a(x)$ denote $(1+x)^r$. Then the derivative of $a(x)$ is $a'(x) = r(1+x)^{r-1}$, the second derivative $a''(x)$ is $r(r-1)(1+x)^{r-2}$, and so on. Generally,

$$a^{(i)}(x) = \left(\prod_{j=0}^{i-1} (r-j) \right) (1+x)^{r-i}.$$

If we evaluate the function at $x = 0$, we get

$$a^{(i)}(0) = \prod_{j=0}^{i-1} (r-j)$$

so we can use Taylor's theorem to get

$$\begin{aligned} a(x) &= \sum_{i \geq 0} \frac{a^{(i)}(0)}{i!} x^i \\ &= \sum_{i \geq 0} \binom{r}{i} x^i, \end{aligned}$$

which is what we wanted. ∎

2.3. Existence of a closed form

It is clear from the theorem above that the generating function for the sequence $\binom{r}{1}, \binom{r}{2}, \ldots$ is $(1+x)^r$. We may wonder what kinds of sequences have a *closed form* generating function. The following theorem, presented without proof, answers this question.

Theorem. *Let $(a_i)_{i=0}^{\infty}$ be a sequence of reals such that there exists some $k > 0$ for which $|a_i| \leq k^i$ for every $i \geq 1$. Then for all x in the interval $(-1/k, 1/k)$, the power series*

$$a(x) = \sum_{i=0}^{\infty} a_i x^i$$

converges, so the generating function $a(x)$ has a closed form. Moreover, for any arbitrarily small $\epsilon > 0$, the values of $a(x)$ for x in the interval $(-\epsilon, \epsilon)$ uniquely determine the sequence $(a_i)_{i=0}^{\infty}$ where, by Taylor's theorem, $a_i = a^{(i)}(0)/i!$ ∎

2.4. Application to counting

Generating functions give us a method for counting the number of elements with some parameter i. For example, we could count the number of trees with i vertices or the number of steps performed by an algorithm with input size i. Often, a_i is defined inductively or recursively.

For example, suppose that for some number c our sequence a_i is given by $a_0 = c$, and $a_i = a_{i-1} + c$ for $i \geq 1$. Then we get that

$$\begin{aligned} a(x) &= \sum_{i \geq 0} a_i x^i \\ &= cx^0 + \sum_{i \geq 1} (a_{i-1} + c) x^i \\ &= cx^0 + \sum_{i \geq 1} cx^i + \sum_{i \geq 1} a_{i-1} \\ &= \frac{c}{1-x} + x \cdot a(x). \end{aligned}$$

This implies that $a(x) = c/(1-x)$, and by Taylor's theorem, $a_i = a^{(0)}(0)/i!$. Calculating this directly can be very tedious, so very often, we will express $a(x)$ in terms of generating functions that we know. In the case of this example, we note that

$$\begin{aligned}\frac{c}{(1-x)^2} &= c \cdot \frac{1}{1-x} \cdot \frac{1}{1-x} \\ &= c\left(\sum_{i\geq 0} x^i\right)\left(\sum_{i\geq 0} x^i\right) \\ &= \sum_{i\geq 0} c(i+1)x^i,\end{aligned}$$

which gives us that $a_i = c(i+1)$ (which we could have easily gotten by observing the recurrence).

2.5. Operations on generating functions

Some useful operations on generating functions include:

Addition. $a(x) + b(x) = \sum_{i\geq 0}(a_i + b_i)x^i$.

Multiplication. $\beta a(x) = \sum_{i\geq 0}(\beta a_i)x^i$.

Scaling. $a(\beta x) = \sum_{i\geq 0}(\beta^i a_i)x^i$.

Gaps. $a(x^k) = \sum_{i\geq 0} a_i x^{ki}$.

Shift right. $xa(x) = \sum_{i\geq 0}(a_{i-1})x^i$.

Shift left. $(a(x) - a_0)/x = \sum_{i\geq 0}(a_{i+1})x^i$.

Derivative. $a'(x) = \sum_{i\geq 0}\big((a+1)a_{i+1}\big)x^i$.

Integral. $\int_0^x a(x)dx = \sum_{i\geq 1}(a_{i-1}/i)x^i$.

Convolution. $a(x)b(x) = \sum_{i\geq 0}\left(\sum_{k=0}^{i} a_k b_{i-k}\right)x^i$.

Partial sums. $a(x)/(1-x) = \sum_{i\geq 0}\left(\sum_{k\geq 0} a_k\right)x^i$.

2.6. Some applications

Imagine a situation at a grocery store. A customer has six \$1 coins, five \$2 coins, and four \$5 bills. Suppose she has to pay \$21. How many ways can she do this? To find the answer, we need only compute the coefficient of x^{21} in the polynomial

$$(1 + x + x^2 + \cdots + x^6) \cdot (1 + x^2 + x^4 + \cdots + x^{10}) \cdot (1 + x^5 + x^{10} + \cdots + x^{20}).$$

Now suppose that a box contains 30 red balls, 40 blue balls, and 50 white balls. In how many ways can one draw 70 balls? As you may have suspected, we need to get the coefficient of x^{70} in

$$(1 + x + \cdots + x^{30}) \cdot (1 + x + \cdots + x^{40}) \cdot (1 + x + \cdots + x^{50}).$$

It may be instructive to work through a full example. We will use generating functions to count binary trees. A *binary tree* T is either empty ($T = \emptyset$) or consists of a root adjacent to a left subtree T_l and a right subtree T_r. Let b_n denote the number of binary trees with n vertices. We will try to find a closed formula for b_n. There is exactly one binary tree with 0 vertices (the empty one), and one binary tree with 1 vertex. From the definition, $n = 0$ or $n = 1 + n_l + n_r$ where $n_l = k$ and $n_r = n - 1 - k$ for some $k \in \{0, \dots, n-1\}$. Hence we obtain the recurrence $b_n = b_0 \cdot b_{n-1} + b_1 \cdot b_{n-2} + \cdots b_{n-1} \cdot b_0$ for $n \geq 1$, i.e.

$$b_n = \sum_{k=0}^{n-1} b_k \cdot b_{n-1-k}.$$

Then the generating function $b(x)$ can be derived:

$$\begin{aligned} b(x) &= \sum_{n \geq 0} b_n x^n \\ &= 1x^0 + \sum_{n \geq 1} \left(\sum_{k \geq 0}^{n-1} b_k \cdot b_{n-1-k} \right) x^n \\ &= 1 + x \sum_{n \geq 0} \left(\sum_{k \geq 0}^{n-1} b_k \cdot b_{n-1-k} \right) x^n \\ &= 1 + x b(x) b(x). \end{aligned}$$

This implies that $xb(x)^2 - b(x) + 1 = 0$, which means that

$$b(x) = \frac{1 \pm \sqrt{1 - 4x}}{2x}$$

Which one is correct? Well, note that $b(0) = \sum_{n \geq 0} b_n 0^n = b_0 = 1$, whereas

$$\lim_{x \to 0} \frac{1 + \sqrt{1 - 4x}}{2x} = +\infty.$$

We try the other root, starting with

$$\lim_{x \to 0} \frac{1 - \sqrt{1 - 4x}}{2x},$$

upon which we can apply l'Hospital's rule to get a limit of 1. So this is the generating function we want.

Applying Newton's formula and the scaling operation, we observe that

$$\sqrt{1 - 4x} = \big(1 + (-4x)\big)^{1/2} = \sum_{n \geq 0} (-4)^n \binom{1/2}{n} x^n.$$

We can substitute this into our generating function to get that

$$b(x) = \frac{1}{2x} \left(1 - \sum_{n \geq 0} (-4)^n \binom{1/2}{n} x^n \right).$$

Since $1 = \sum_{n \geq 0} (-4)^n \binom{1/2}{n} x^0$, we can further simplify, obtaining

$$\begin{aligned} b(x) &= \frac{1}{x} \sum_{n \geq 1} -\frac{1}{2} (-4)^n \binom{1/2}{n} x^n \\ &= \sum_{n \geq 0} -\frac{1}{2} (-4)^{n+1} \binom{1/2}{n+1} x^n \\ &= \sum_{n \geq 0} \frac{1}{n+1} \binom{2n}{n} x^n. \end{aligned}$$

So the number of binary trees with n vertices is $\binom{2n}{n}/(n+1)$. These are the *Catalan numbers*.

3. GRAPHS

3.1. Network flow

3.1.1. Edge capacities

Suppose we are given a directed graph $G = (V, E)$ ($E \subseteq V \times V$) with designated source $s \in V$ and target $t \in V$. To each edge e is assigned a capacity $c(e)$, which may be a non-negative real number or $+\infty$. Then an *s-t-flow* is a function $f : E \to \mathbf{R}$ such that for every edge $e \in E$, $0 \leq f(e) \leq c(e)$. These are called the *capacity constraints*. Our flow must also obey the law of *flow conservation*, also known as *Kirchhoff's First Law* (which was originally observed in electrical circuits): For every vertex $v \in V \setminus \{s, t\}$, $\sum_{uv \in E} f(uv) = \sum_{vu \in E} f(vu)$. We want to maximise the flow that leaves s, subject to these constraints. The *value* of an s-t-flow is

$$|f| = \sum_{su \in E} f(su) - f(us),$$

and a *max-flow* is an s-t-flow with maximum value.

The intuitive way to check that a flow is maximal is to study "bottlenecks". Concretely, if we let $S \subseteq V$ be such that $s \in S$ and $t \notin S$, then the set $\delta^+(S) = \{uv \in E : u \in S, v \notin S\}$ is an *s-t-cut* or *s-t-edge-cut*. After removing $\delta^+(S)$ from G, there is no longer a path from s to t, i.e. $\delta^+(S)$ *separates* s from t. (Since we only removed edges *leaving* S, there may still be a path from t to s.) The *capacity* of an s-t-cut is given by

$$c\big(\delta^+(S)\big) = \sum_{uv \in \delta^+(S)} c(uv).$$

Then we can define a *min-cut* to be an s-t-cut with minimum capacity.

Lemma. *For any s-t-flow f and s-t-cut $\delta^+(S)$,*

$$|f| = \sum_{e \in \delta^+(S)} f(e) - \sum_{e \in \delta^-(S)} f(e),$$

where $\delta^-(S)$ is the set $\{uv : u \notin S, v \in S\}$.

Proof. By flow conservation, we know that for every $v \neq s, t$,

$$\sum_{vu \in E} f(vu) - \sum_{uv \in E} f(uv) = 0.$$

Then the value of the s-t-flow is the sum of all equations for the cut S:

$$|f| = \sum_{v \in S} \left(\sum_{vu \in E} f(vu) - \sum_{uv \in E} f(uv) \right).$$

We need to consider three possible cases for an edge $e = uv$:

1. ($u, v \in S$) If both u and v are in S, then we will add $f(uv)$ and then subtract $f(uv)$, so this case does not contribute to $|f|$.
2. ($u \in S$, $v \notin S$) In this case, $uv \in \delta^+(S)$, so we will have a contribution of $+f(uv)$ to $|f|$.
3. ($u \notin S$, $v \in S$) In this case, $uv \in \delta^-(S)$, so we will have a contribution of $-f(uv)$ to $|f|$.

This implies that

$$|f| = \sum_{e \in \delta^+(S)} f(e) - \sum_{e \in \delta^-(S)} f(e),$$

which is exactly what we wanted to prove. ∎

We can use this to derive an easy corollary:

Corollary. *If f is a max-flow and $\delta^+(S)$ is a min-cut, then $|f| \leq c\big(\delta^+(S)\big)$.*

Proof. By the preceding lemma, $|f| \leq \sum_{e\in\delta^+(S)} f(e)$, which by capacity constraints is no greater than $\sum_{e\in\delta^+(S)} c(e)$. But by the definition of the capacity of a min-cut, this is just $c\big(\delta^+(S)\big)$. ∎

In fact, the value of a max-flow *equals* the capacity of a min-cut, but the other inequality requires a more involved proof.

Theorem. *If f is a max-flow and $\delta^+(S)$ is a min-cut, then $|f| = c\big(\delta^+(S)\big)$.*

Proof. By the preceding corollary, it suffices to show that $|f| \geq c\big(\delta^+(S)\big)$. For any max-flow f, the idea is to construct some s-t-cut $\delta^+(S)$ with $|f| = \delta^+(S)$. This would imply that any *min-cut* would have capacity no greater than $|f|$.

Consider the following procedure. We start with $S \leftarrow \{s\}$. While there exists an edge $uv \in \delta^+(S)$ such that $c(uv) > f(uv)$ *or* there exists an edge $vu \in \delta^-(S)$ such that $f(uv) > 0$, we add v to S. The claim is that if f is a max-flow, then this algorithm will produce an S such that $t \notin S$, i.e. S is an s-t-cut. If this claim is true, then we are done; since $\sum_{e\in\delta^-(S)} = 0$ after the algorithm terminates, we will have $|f| = c\big(\delta^+(S)\big)$.

Suppose, towards a contradiction, that $t \in S$. Then there exists a sequence of vertices $v_0, \ldots, v_k \in S$ such that $v_0 = s$, $v_k = t$, and for all $i \in \{0, \ldots, k-1\}$, either

a) $v_iv_{i+1} \in E$ and $c(v_iv_{i+1}) > f(v_iv_{i+1})$, or
b) $v_{i+1}v_i \in E$ and $f(v_{i+1}v_i) > 0$.

For each i, define a small real number ϵ_i given by

$$\epsilon_i = \begin{cases} c(v_iv_{i+1}) - f(v_iv_{i+1}) & \text{if case (a) holds} \\ f(v_{i+1}) & \text{if case (b) holds} \end{cases}$$

Then we can let $\epsilon = \min\{e_i : 0 \leq i \leq k-1\}$. Note that, by construction, $\epsilon > 0$.

Now we create a new function $f' : E \to \mathbf{R}$ given by

$$f'(e) = \begin{cases} f(e) + \epsilon & \text{if e satisfies case (a)} \\ f(e) - \epsilon & \text{if e satisfies case (b)} \\ f(e) & \text{otherwise} \end{cases}$$

The claim is that f' is an s-t-flow. The values of $f'(e)$ are non negative, because $f'(e) \geq f(e) \geq 0$ in case a), and in case b), $f'(e) = f(e) - \epsilon \geq 0$. The capacity constraints are satisfied, since in case a), $\epsilon \leq c(e) - f(e)$ implies that $f'(e) = f(e) + \epsilon \leq c(e)$; and in case b), $f'(e) \leq f(e) \leq c(e)$. And we can see that the law of flow conservation is obeyed as well, since for every $v \in V \setminus \{s, t\}$, either flow is unchanged at v or exactly two edges are changed (one adds ϵ and the other one subtracts ϵ from f).

Computing the value of the flow f', we find that

$$|f'| = \sum_{sv\in E} f'(sv) - \sum_{vs\in E} f'(vs) = |f| + \epsilon.$$

This implies that f is not a max-flow, a contradiction. ∎

So to determine if a flow is maximal, we can figure out what S is and see if $t \in S$. If not, then f is a max-flow and $\delta^+(S)$ is a min-cut. The proof of the theorem also suggests an algorithm to compute max-flow and min-cut:

Algorithm M (*Find max-flow/min-cut*). Given a directed graph $G = (V, E)$, this algorithm finds a max-flow f and a set $S \subset V$ such that $\delta^+(S)$ is a min-cut.

M1. [Initialise.] Set $i \leftarrow 0$ and $f_0(e) \leftarrow 0$ for every $e \in E$.

M2. [Find S_i] Construct the set S_i for the flow f_i using the procedure outlined in the preceding proof.

M3. [Is t in S_i?] If $t \notin S_i$, the flow f_i is maximal. Output f_i and S_i.

M4. [Increase flow.] Find the "path" P from s to t in S, and compute ϵ. Let f_{i+1} be the result of augmenting f_i by ϵ along P. Set $i \leftarrow i + 1$.

M5. [Repeat.] Go to step M2. ∎

It is clear that if the algorithm terminates, it will output the correct answer. So it is natural to wonder when the algorithm is sure to terminate. In the case of integer capacities, termination is easy to prove:

Theorem. *If for all $e \in E$, $c(e) \in \mathbf{N}_0$, then the capacity n of the min-cut is integral and Algorithm M terminates in at most n iterations.*

Proof. We prove, by induction on i, that at every step of the algorithm, $|f_i| \in \mathbf{N}_0$. The base case is simple because the algorithm starts with $|f_0| = 0$. Now assume that $f_i \in \mathbf{N}_0$. In the step that increases i to $i+1$, we choose ϵ to be the minimum value computed over a path P of edges. So for some $e \in P$, either $\epsilon = c(e) - f_i(e)$ or $\epsilon = f_i(e)$. Since both $c(e), f_i(e) \in \mathbf{N}_0$, we have that $\epsilon \in \mathbf{N}_0$ and the value of the flow at each iteration is integral.

Then because $\epsilon > 0$, we have from integrality that $\epsilon \geq 1$ meaning that for every iteration of the algorithm, the value $|f_{i+1}| = |f_i| + \epsilon \geq |f_i| + 1$. Since the value of the max-flow is exactly n, the algorithm terminates in at most n iterations. ∎

3.1.2. Vertex capacites

Instead of constraining flow at the edges, we can instead assign capacities $d(v)$ to vertices $v \in V \setminus \{s,t\}$. Then for all $v \in V \setminus \{s,t\}$, an s-t-flow $f : E \to \mathbf{R}$ must satisfy $\sum_{uv \in E} f(uv) \leq d(v)$ and $\sum_{uv \in E} f(uv) = \sum_{vu \in E} f(vu)$. Note that by this definition, if $st \in E$, then the value of the max-flow will be infinite, so assume that $st \notin E$.

Now we define an *s-t-vertex-cut* to be a set $T \subseteq V \setminus \{s,t\}$ such that the graph $G \setminus T$ has no path from s to t. For ease of notation, let $d(T) = \sum_{v \in T} d(v)$.

Theorem. *Let f be a max-flow in a network with vertex capacities. If T is a min-cut, then $|f| = d(T)$, and furthermore, if all capacities are integers, then there is an integer max-flow.*

Proof. We simply convert our vertex-constrained network to an edge-constrained one. Replace each constrained vertex v_i with two new vertices v_{i1} and v_{i2}. Connect these vertices with a single edge e_i with $c(e_i) = d(v_i)$. Set the capacties of all other edges in the new graph to $+\infty$. Then the theorem follows from our previous theorems regarding edge-constrained networks. ∎

3.2. Bipartite matchings

A graph $G = (V, E)$ is *bipartite* if V can be split into disjoint subsets A and B such that every edge in E has one endpoint in A and the other in B. A *matching* is a set $M \subseteq E$ such that no two edges of M share a vertex. A *maximum matching* is matching of maximum size.

Consider the so-called *marriage problem*. Given a set A of boys and B of girls, can we marry off all boys to girls that they know? We can model this problem with a bipartite graph $(A \cup B, E)$, where an edge exists between a boy and a girl if they know each other. First we define the set of *neigbours* of a vertex set $S \subseteq A$ to be $N(S) = \{v \in B : uv \in E \text{ and } u \in S\}$. Then the following theorem tells us exactly when the marriage problem is solvable.

Theorem (*Hall*). *A bipartite graph $G = (A \cup B, E)$ (with A and B disjoint) has a matching $M \subseteq E$ of size $|M| = |A|$ if and only if $|N(S)| \geq |S|$ for all $S \subseteq A$.*

Proof. The "only if" direction is obvious. We prove the "if" direction by contrapositive, i.e. if there is no matching of size $|A|$ then there exists some $S \subseteq A$ such that $|N(S) < |S|$.

The idea is to use the vertex version of the max-flow min-cut theorem. We construct a directed graph $H = (V', E')$ from G. Introduce two new vertices s and t and set $V' = A \cup B \cup \{s,t\}$. We want to connect s to every vertex in A and connect every vertex in B to t. Then for every edge already in the graph, ensure it is directed from A to B. So

$$E' = \{su : u \in A\} \cup \{vt : v \in B\} \cup \{uv : uv \in E, u \in A, v \in B\}.$$

Now we set all the capacities on every vertex $v \in A \cup B$ to 1. We know from the max-flow min-cut theorem that there exists an integer max-flow, call it f, which is a *0-1 flow*, meaning that $f(e) \in \{0,1\}$ for all $e \in E'$. Note that if $f \geq |A|$, then there exists a matching of size $|A|$, since the 0-1 flow uses at most one incoming and one outgoing edge of any vertex in $V \setminus \{s,t\}$.

But we assumed that no matching of size $|A|$ exists, so $|f| < |A|$. Let T be a min-cut. From the max-flow min-cut theorem, we know that $d(T) = |f| < |A|$. Consider the sets $X = A \cap T$ and $Y = B \cap T$. Since $|X| + |Y| = |T| = \sum_{v \in T} 1 = d(T)$, we have that $|Y| < |A| - |X|$. There is no edge from $A \setminus X$ to $B \setminus Y$, as T is an s-t-vertex-cut. So $N(A \setminus X) \subseteq Y$ in G and thus $|N(A \setminus X)| \leq |Y| < |A| - |X| = |A \setminus X|$. The set $A \setminus X$ is exactly the S we were looking for. ∎

In a graph $G = (V, E)$, a *vertex cover* is a set $C \subseteq V$ such that every edge is incident to at least one vertex in C. A *minimum vertex cover* is a vertex cover of minimum cardinality. Notice that every vertex cover of G is an s-t-vertex cut in H, otherwise there would be a path from s to t. So we can derive the following theorem as a corollary.

Theorem (*Kőnig's theorem*). *Let G be a bipartite graph with maximum matching M and minimum vertex cover C. Then $|M| = |C|$.* ∎

3.3. Graph connectivity

3.3.1. Definitions and inequalities

A graph $G = (V, E)$ is *connected* if there is a path between any two vertices. Otherwise, we say G is *disconnected*. A *component* of G is a maximal connected subgraph. If G is connected and removing a set W of vertices or edges causes it to become disconnected, then we say W *separates* G. We have special names for W if $|W| = 1$. If such a $W \subseteq V$, we call it a *cut vertex*. If $W \subseteq E, |W| = 1$, we call W a *bridge*.

A graph $G = (V, E)$ is *k-connected* if $|V| > k$ and no vertex set $W \subseteq V$ of size $|W| < k$ separates G. Note that the complete graph K_n is $(n-1)$-connected and that if G is k-connected then it is also k'-connected for any $k' \leq k$. If $|V| \geq 2$ and no edge set $W \subseteq E$ of size $|W| < k$ separates G, then G is *k-edge-connected*.

The *connectivity* of a graph G, denoted $\kappa(G)$, is the maximum $k \in \mathbf{N}$ such that G is k-connected. Likewise, the *edge-connectivity* $\lambda(G)$ of a graph G is the maximum k such that G is k-edge-connected. Note that if G is not a complete graph, then both $\kappa(G)$ and $\lambda(G)$ are at most $|V| - 2$.

Lemma. *Let $G = (V, E)$ be a graph. Then for all edges $uv \in E$, $\kappa(G \setminus \{uv\}) \geq \kappa(G) - 1$.*

Proof. Remove an edge uv from G. Now we find a set of vertices W, with $|W| = \kappa(G \setminus \{uv\})$, that disconnects G. So V has been split into three disjoint subsets $L \cup W \cup R$. If u and v both lie in L or both lie in R, then $\kappa(G) \leq |W| = \kappa(G \setminus \{uv\})$. If $u \in L$ and $v \in R$ (or vice-versa), then if there exists a vertex $w \neq u \in L$, then $W \cup \{u\}$ separates G, implying that $\kappa(G) \leq |W \cup \{u\}| \leq \kappa(G \setminus \{u, v\}) + 1$. The last case is if $L = \{u\}$ and $R = \{v\}$. Then $|V| = |W \cup \{u, v\}| \leq \kappa(G \setminus \{u, v\} + 2$, meaning that $\kappa(G) \leq |V| - 1 \leq \kappa(G \setminus \{u, v\}) + 1$. ∎

Lemma. *Let $G = (V, E)$ be a graph with $|V| = n$. Then the following both hold:*

1. *$\kappa(G) - 1 \leq \kappa(G - v)$ for all $v \in V$.*
2. *$\lambda(G) - 1 \leq \lambda(G - e) \leq \lambda(G)$ for all $e \in E$.*

Proof. We prove each part separately.

1. If $G = K_n$, then $G - v = K_{n-1}$, so $\kappa(G) - 1 = n - 2 = \kappa(G - v)$ and we are done. So assume $G \neq K_n$. If $G \neq K_n$ and $G - v = K_{n-1}$, then there must exist some vertex $u \in V$ such that $uv \notin E$. Then removing the set $W = V \setminus \{u, v\}$ separates G, so $\kappa(G) \leq |W| = n - 2 = \kappa(G - v)$. The last case is that $G \neq K_n$ and $G - v \neq K_{n-1}$. Then there exists a W with $|W| = \kappa(G - v)$ such that $W \cup \{v\}$ separates G, which means that $\kappa(G) \leq |W \cup \{v\}| = \kappa(G - v) + 1$.
2. There exists a set $W \subseteq E$ with $|W| = \lambda(G - e)$. So $e \notin W$ but $W \cup \{e\}$ separates G. So $\lambda(G) \leq |W \cup \{e\}| = \lambda(G - e) + 1$. For the second inequality, let W be the set that separates G, so $|W| = \lambda(G)$. Then $W \setminus \{e\}$ separates $G - e$. So $\lambda(G - e) \leq |W \setminus \{e\}| \leq |W| = \lambda(G)$. ∎

To summarise, removing an edge from G causes both $\kappa(G)$, $\lambda(G)$ to decrease by *at most* 1. Removing a vertex from G can cause $\kappa(G)$ to decrease by at most 1 as well, but may cause $\lambda(G)$ to decrease a lot (possibly down to 0). The next lemma describes the relationship between vertex- and edge-connectivity.

Lemma. *Let $G = (V, E)$ be a graph and let $\delta(G)$ denote the minimum degree over all vertices in G. Then*

$$\kappa(G) \leq \lambda(G) \leq \delta(G).$$

Proof. The inequality $\lambda(G) \leq \delta(G)$ is easy, because if v is the vertex of minimum degree in G, then removing the $\delta(G)$ edges adjacent to v separates v from the rest of the graph.

To show that $\kappa(G) \leq \lambda(G)$, we use that $\kappa(G - e) \geq \kappa(G) - 1$ for all $e \in E$. Suppose $\lambda(G) = n$. So removing a set of edges $\{e_1, \ldots, e_n\}$ causes G to be disconnected. This means that $0 = \kappa(G - \{e_1, \ldots, e_n\}) \geq \kappa(G) - n$, which means that $\kappa(G) \leq n = \lambda(G)$. ∎

3.3.2. Connectivity and paths

For the following important theorem, we will need a new definition. We say two s-t-paths P and Q are *independent* if $V(P) \cap V(Q) = \{s, t\}$. The paths are *edge-disjoint* if $E(P) \cap E(Q) = \emptyset$.

Theorem M (*Menger*). *Let G be a graph. The following both hold:*

1. *G is k-connected if and only if there are k independent paths between any two vertices s and t.*
2. *G is k-edge-connected if and only if there are k edge-disjoint paths between any two vertices s and t.*

Proof. The "if" direction is left as an exercise for the student. To prove the "only if" direction, we use the max-flow min-cut theorem. We construct a directed graph G' from G:

$$V(G') = V(G)$$
$$E(G') = \{uv, vu : uv \in E(G)\}$$

To prove the vertex version of the theorem, for every vertex $v \in V' \setminus \{s, t\}$, we set a vertex capacity $d(v) = 1$. To prove the edge version of the theorem, we set $c(e) = 1$ for all edges in $E(G')$. Now we consider the vertex and edge cases separately.

1. Assume that $st \notin E(G')$. Any s-t-vertex cut of G' separates G; since any s-t-path in G' is also a t-s-path, the s-t-vertex cut also separates t from s. Let T denote a minimum-cardinality s-t-vertex-cut in G' and we get that $d(T) \geq \kappa(G)$. There exists a 0-1 flow with $|f| \geq \kappa(G)$. So at each vertex, the number of incoming edges is at most 1 and the number of outgoing edges is at most 1 as well. This implies that there are $|f|$ independent paths from s to t. But by the max-flow min-cut theorem, $|f| = d(T) \geq \kappa(G) = k$. If it so happens that $st \in E(G)$, then (s, st, t) is an additional independent s-t-path to those in $G - st$ and we know that $\kappa(G - st) \geq \kappa(G) - 1$. Hence G being k-connected implies that there are at least k independent paths between any s, t.
2. In the case of edge capacities, any s-t-edge-cut separates G as any t-s-path is also an s-t-path in G'. Let S denote a minimum s-t-edge cut. Then $c(E) \geq \lambda(G)$. Again, there exists a 0-1 flow f with $|f| \geq \lambda(G)$, but this time it may be possible that a vertex has many incoming and outgoing edges. However, we know that the number of incoming edges that carry flow must equal the number of outgoing edges that carry flow at any vertex. Let $H = (V, F)$ be a subgraph of G, where $F = \{e \in E(G) : f(e) = 1\}$. While there exists an s-t-path P in H, we can remove any all edges of P from H while maintaining flow conservation at every vertex $v \in V \setminus \{s, t\}$. We can repeat the algorithm as many times as there are edges coming out of S, so at least $|f|$ times. Hence $|f| \geq \lambda(G) \geq k$. ∎

3.3.3. 2-connectivity and ear decompositions

An *ear decomposition* of a graph G is a sequence $G_0, G_1, \ldots, G_k$ of subgraphs of G such that G_0 is a cycle and for any $i \in \{1, \ldots, k\}$, the graph G_i is obtained from G_{i-1} by adding a path P_i to G_{i-1} such that P_i shares exactly its endpoints with G_{i-1}. Each of these paths P_i is called an *ear*.

Theorem. *A graph is 2-connected if and only if it has an ear decomposition.*

Proof. The only way for a graph to not be 2-connected is for it to have a cut vertex. So the "if" direction is easy, because if there is an ear decomposition, there is no cut vertex.

To prove the "only if" direction, suppose that G is a 2-connected graph, and let G_0 be an arbitrary cycle in G. To get from any subgraph G_{i-1} to G_i, we need to find an ear. Since G is connected and $G_{i-1} \neq G$, there exists an edge $uv \in E(G) \setminus E(G_{i-1})$ such that $u \in V(G_{i-1})$. If $v \in V(G_{i-1})$, then uv is an ear and we can set $P_i = uv$. Otherwise, there exists an edge $uw \in E(G_{i-1})$. This we know because $G_0 \subseteq G_{i-1}$ so $|V(G_{i-1})| \geq 3$ and G_{i-1} is connected.

By Menger's theorem, there exist at least 2 independent paths between v and w in G. So there exists some v-w-path P such that $u \notin V(P)$. Let u' be the first vertex of P that is in G_{i-1}. Then we can append u to the front of P to get a path $(u, uv, v, \ldots, u')$, which is an ear. ∎

For a graph G, we define an *edge addition* is the operation of adding an edge between two vertices that were not connected in G. An *edge subdivision* is the operation of splitting an edge into two edges, with a new vertex in between them.

Corollary. *A graph is 2-connected if it can be obtained from K_3 by a sequence of edge additions and subdivisions.*

This corollary is immediate from the observation that any starting cycle G_0 is a subdivision of K_3, and any ear P_i is a subdivision of an edge addition.

3.4. Counting spanning trees

A *spanning tree* T of a graph G is a subgraph that contains all vertices in G and is a tree with edges from G. Let $T(G)$ denote the number of spanning trees of a graph G. For example, $T(K_3) = 3$. We want to find out the value of $T(G)$ for different classes of graphs.

In our investigation of $T(K_n)$, we will need to define a certain type of finite sequence. A Prüfer code is a sequence $(p_1, \ldots, p_{n-2}) \in \{1, \ldots, n\}^{n-2}$. It turns out that there is a bijection between Prüfer codes and trees of length $n-2$ and trees on n vertices. First we consider the algorithm for getting the Prüfer code from a tree on n vertices:

Algorithm P (*Build Prüfer code*). Given a tree with n vertices T, this algorithm constructs a Prüfer code.

P1. [Initialise.] Set $i \leftarrow 1$, $T_0 \leftarrow T$.

P2. [Get next element.] Let l_i denote the leaf of T_{i-1} with smallest index. Set p_i equal to the (single) neighbour of l_i in T_{i-1}.

P3. [Remove a leaf.] Set $T_i \leftarrow T_{i-1} - l_i$.

P4. [Done?] If $i = n-2$, we output the sequence $(p_1, \ldots, p_{n-2})$.

P5. [Repeat.] Otherwise, set $i \leftarrow i+1$ and return to step P2. ∎

There is a reverse algorithm that builds a tree from a Prufer code:

Algorithm T (*Build a tree*). Given a Prüfer code $(p_1, \ldots, p_{n-2})$, this algorithm outputs a tree with n labelled vertices.

T1. [Calculate l_i.] For every $i \in \{1, \ldots, n\}$, let l_i denote the number of times i appears in the Prüfer code, plus 1.

T2. [Initialise.] Let $T_0 \leftarrow (\emptyset, \emptyset)$ to begin with. Set $i \leftarrow 1$.

T3. [Add an edge.] Let j be the smallest index with $l_j = 1$. Set $T_i \leftarrow T_{i-1} + \{p_i, j\}$, where the addition operation denotes adding the edge as well as the vertices p_i, j, if one (or both) of them is not yet in the tree.

T4. [Update l.] Set $l_{p_i} \leftarrow l_{p_i} - 1$ and $l_j \leftarrow l_j - 1$.

T5. [Done?] If $i = n-2$, we have worked through the whole Prüfer code. There are two remaining non-zero elements in l, call them l_x and l_y. Output $T_i + \{l_x, l_y\}$.

T6. [Repeat.] Otherwise, set $i \leftarrow i+1$ and return to step T3. ∎

Since Algorithms P and T are both deterministic and they are inverse to one another, this implies a bijection between trees on n vertices and Prüfer codes of length $n-2$. Hence we have the following formula.

Theorem C (*Cayley's formula*). *The number of spanning trees of a complete graph with n vertices is*

$$T(K_n) = n^{n-2}.$$

Proof. Since every possible edge between two vertices exists in K_n, the number of spanning trees of K_n is exactly the number of trees with n vertices. This is the same value as the number of Prüfer codes of length $n-2$, which is the value n^{n-2}, since there are n choices for each of $n-2$ elements. ∎

The following theorem gives a formula for the number of spanning trees of K_n, after a single edge is removed.

Theorem. *Let e be an edge in K_n. Then*

$$T(K_n - e) = (n-2)n^{n-3}.$$

Proof. The idea is to double-count the number of edges in all spanning trees of K_n. Each spanning tree has $n-1$ edges so the total number of edges is $(n-1)n^{n-2}$. Now let k_e denote the number of spanning trees of K_n containing e. By symmetry, this is the same for any choice of e in K_n. There are $\binom{n}{2}$ edges in K_n, each contained in k_e trees. This implies that

$$\binom{n}{2} k_e = (n-1)n^{n-2}.$$

Solving for k_e, we obtain the value $k_e = 2n^{n-3}$. Therefore $T(K_n - e) = T(K_n) - k_e = n^{n-2} - 2n^{n-3} = (n-2)n^{n-3}$, which is what we wanted. ∎

4. EXTREMAL THEORY

4.1. Extremal graphs

This section deals with the general question: "What is the extremal (maximum/minimum) number of objects, subject to restriction R?" Some simple examples from graph theory include

- Question: What is the maximum number of edges in a graph with n vertices? Answer: $\binom{n}{2}$.
- Question: What is the maximum number of edges in a graph with no cycles? Answer: $n-1$.

A harder question is "What is the maximum number of edges in a graph not containing C_3 as a subgraph?" First let us consider a class of graphs that definitely do not contain C_3 as a subgraph: bipartite graphs (they do not contain any odd cycles). What is the maximum number of edges in a bipartite graph? Well, the complete bipartite graph $K_{a,b}$ has ab edges, so we want to maximise the product ab when $a+b=n$. Intuitively this is when a and b are as close to $n/2$ as possible. So we get that the number of edges in $K_{a,b}$ is at least $\lfloor n^2/4 \rfloor$, thus we have a lower bound for the number of edges in a graph not containing C_3. The following theorem proves that the bound is tight.

Theorem. *The maximum number of edges of any graph G not containing C_3 as a subgraph is at most $\lfloor n^2/4 \rfloor$.*

Proof. We want to show that there exist $a, b \in \mathbf{N}$ such that $|E(G)| \leq |E(K_{a,b})| \leq \lfloor n^2/4 \rfloor$. To this end, we find disjoint subsets $A, B \subseteq V(G)$ such that $A \cup B = V(G)$. and let $H = K_{a,b}$ on A and B.

Then it suffices to show that for any $v \in V(G)$, $\deg_G(v) \leq \deg_H(v)$, since the number of edges in any graph equals the sum over degrees of vertices, divided by 2. So let v_0 be a vertex of maximum degree in G. Then we can set $B = N_G(v_0)$ and $A = V \setminus B$ Now for any $v \in A$, $\deg_H(v) = |B| = \deg_G(v_0) \geq \deg_G(v)$. Note that in G, no two vertices of B are adjacent, since then there would be a cycle of length 3. Then for any $v \in B$, $N_G(v) \subseteq A$, hence $\deg_G(v) \leq |A| = \deg_H(v)$. ∎

Compared to $\binom{n}{2}$, $\lfloor n^2/4 \rfloor$ is still quite a lot, but remember that disallowing any cycles at all, we have a much smaller bound of $n-1$. A natural question is "How long do forbidden cycles have to be so that the maximum number of edges is much smaller than $\binom{n}{2}$?" As we will see from the following theorem, disallowing C_4 lowers this bound quite a bit.

Theorem. *The maximum number of edges of any graph G not containing C_4 as a subgraph is at most $(n^{3/2}+n)/2$.*

Proof. The idea is to double-count the size of the set M of pairs $(\{u,u'\},v)$, where $u \neq u' \neq v$ and $uv, u'v \in E$. Note that for any set $\{u,u'\}$, there is at most one $v \in V$ such that $(\{u,u'\},v) \in M$, because otherwise we would have a cycle of length 4, which is forbidden. This implies that $|M| \leq \binom{n}{2}$, which is the number of sets $\{u,u'\}$.

Another way to count $|M|$ is to consider the number of sets $\{u,u'\}$ that are contributed to by each $v \in V$. For every set $\{u,u'\} \subseteq N(v)$ we get a pair $(\{u,u'\},v) \in M$, so v contributes $\binom{\deg(v)}{2}$ elements to M. For ease of notation, let us number the vertices $V = \{1,\dots,n\}$ and let d_i denote $\deg(i)$. Then

$$|M| = \sum_{i=1}^{n} \binom{d_i}{2} \leq \binom{n}{2}.$$

To get a bound on the number of edges, we will relate $(\sum_{i=1}^{n} d_i)/2$ to $|M|$. We know that $\binom{n}{2} \leq n^2/2$. Without loss of generality, we may assume that $d_i \geq 1$ for all $i \in V$, since adding an edge between a vertex i with $d_i = 0$ to any other vertex j increases the number of edges without introducing a C_4. Therefore,

$$\binom{d_i}{2} = \frac{d_i(d_i-1)}{2} \geq \frac{1}{2}(d_i-1)^2$$

for every $i \in V$. This implies that

$$\sum_{i=1}^{n} (d_i-1)^2 \leq n^2.$$

Now we use the famous *Cauchy-Schwarz inequality*, which applies to vectors $(x_1,\dots,x_n)$, $(y_1,\dots,y_n) \in \mathbf{R}^n$:

$$\sum_{i=1}^{n} x_i y_i \leq \sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}.$$

We apply the formula with $x_i = d_i - 1$ and $y_i = 1$ to obtain

$$\sum_{i=1}^{n} (d_i-1)\cdot 1 \leq \sqrt{\sum_{i=1}^{n} (d_i-1)^2}\sqrt{\sum_{i=1}^{n} 1} \leq \sqrt{n^2}\sqrt{n} = n^{3/2}.$$

Because $\sum_{i=1}^{n}(d_i-1) = \sum_{i=1}^{n} d_i - n$, we get $\sum_{i=1}^{n} d_i \leq n^{3/2}+n$. So we can put a bound on the number of edges:

$$|E| = \frac{1}{2}\sum_{i=1}^{n} d_i \leq \frac{n^{3/2}+n}{2}. \quad \blacksquare$$

4.2. Partially-ordered sets

A *partially-ordered set* or *poset* is a pair $(\mathcal{L},\subseteq)$ where $\subseteq$ is a partial order on the set $\mathcal{L} \subseteq 2^X$ over $X = \{1,\dots,n\}$. A *chain* is a subset $\{A_1,\dots,A_k\}$ of $\mathcal{L}$ such that $A_1 \subseteq A_2 \subseteq \cdots \subseteq A_k$. An *antichain* or *independent set system* is a subset $\{A_1,\dots,A_k\}$ of $\mathcal{L}$ such that $A_i \not\subseteq A_j$ for all $i,j \in \{1,\dots,k\}, i \neq j$. A chain is *maximal* if adding any set breaks the chain property.

The maximum length of a chain in $(\mathcal{L},\subseteq)$ is $|X|+1$, where X denotes the motherset. This is because $\emptyset$ can be part of the chain as well. Now consider the maximum length of an *antichain*. One way to get an antichain is to take all sets $A \in \mathcal{L}$ such that $|A| = i$ for some i. This number is at least $\binom{n}{i} \leq \binom{n}{\lfloor n/2 \rfloor}$ (in the case that $\mathcal{L} = 2^X$). The following theorem gives an upper bound:

Theorem S (*Sperner*). *Any antichain of a poset of $X = \{1, \dots, n\}$ has size at most $\binom{n}{\lfloor n/2 \rfloor}$*

Proof. We will work with $\mathcal{L} = 2^X$. Let an antichain $M \subset 2^X$ be given. Our goal is to bound the cardinality of M. The idea is to double count the number of pairs (R, A) where $A \in M$ and R is a maximal chain containing A. Notice that a maximal chain contains exactly one set of size i for $i \in \{0, \dots, n\}$:

$$\emptyset \subseteq \{x_1\} \subseteq \{x_1, x_2\} \subseteq \cdots \subseteq \{x_1, x_2, \dots, x_n\}.$$

where $x_1 \dots x_n$ are elements of X written in some order. This implies that the number of maximal chains is $n!$ as every possible ordering defines a chain. By observation, any R contains at most one $A \in M$, so the number of pairs (R, A) is at most $n!$.

The other way we will count the pairs is to ask how many maximal chains actually contain a set $A \in M$. If R is made up of elements $x_1, \dots, x_n$ as shown above, then $A \in R$ if and only if $A = \{x_1, \dots, x_k\}$ for some k. Hence to form R we first introduce $x_1, \dots, x_k$ in $k!$ ways, then $x_{k+1}, \dots, x_n$ in $(n-k)!$ ways, so the number of pairs (R, A) is

$$\sum_{A \in M} |A|!(n - |A|)! \leq n!$$

Dividing by $n!$ on both sides of the equation we get that

$$\sum_{A \in M} \frac{|A|!(n - |A|)!}{n!} = \sum_{A \in M} \binom{n}{|A|}^{-1} \leq 1.$$

We know that $\binom{n}{|A|} \leq \binom{n}{\lfloor n/2 \rfloor}$, hence we may replace the reciprocal of the former with the reciprocal of the latter to get

$$1 \geq \sum_{A \in M} \binom{n}{\lfloor n/2 \rfloor}^{-1} = \frac{|M|}{\binom{n}{\lfloor n/2 \rfloor}},$$

which means that $|M| \leq \binom{n}{\lfloor n/2 \rfloor}$. ∎

5. FINITE PROJECTIVE PLANES

Let X be a finite point set and $\mathcal{L} \subseteq 2^X$ be a set of subsets of X. $(X, \mathcal{L})$ is called a *finite projective plane* if the following properties hold:

P0) There exists a set $F \subseteq X$ such that $|F| = 4$ and for all $L \in \mathcal{L}$, $|F \cap L| \leq 2$.

P1) For all $L_1, L_2 \in \mathcal{L}$ with $L_1 \neq L_2$, we have $|L_1 \cap L_2| = 1$.

P2) For all $x_1, x_2 \in X$ with $x_1 \neq x_2$, there is *exactly one* $L \in \mathcal{L}$ such that $x_1, x_2 \in L$ and we may denote it $L = \overline{x_1 x_2}$.

We call any $x \in X$ a *point* and any $L \in \mathcal{L}$ a *line*. Finite projective planes have very interesting "symmetrical" properties.

Lemma. *Let $(X, \mathcal{L})$ be a finite projective plane. Then for all $L, L' \in \mathcal{L}$, there exists an $z \in L$ such that $z \notin L \cup L'$*

Proof. By (P0), there exists a set $F \subseteq X$ with $|F| = 4$ and $|F \cap L| \leq 2$ for a $|F \cap L'| \leq 2$. If F is a proper subset of $L \cup L'$, then we're done, because $z \in F$. So assume $F \subset L \cup L'$: say $F = \{a, b, c, d\}$, $F \cap L = \{a, b\}$, and $F \cap L' = \{c, d\}$. Then by (P2), there exist lines $L_1 = \overline{ac}$ and $L_2 = \overline{bd}$; and by (P1), there exists a point $z \in L_1 \cap L_2$. We claim that $z \notin L \cup L'$.

Suppose, towards a contradiction, that $z \in L$. Then since, by (P1), $|L \cap L_1| = 1$ and $z \in L_1$, we have that $z = a$. This implies that L_2 contains a, b, d. So $|F \cap L_2| \geq 3$, a contradiction to (P0). An analogous argument can be made for L' so we are done. ∎

Lemma. *Let $(X, \mathcal{L})$ be a finite projective plane. Then for all $L, L' \in \mathcal{L}$, $|L| = |L'|$.*

Proof. The idea is to find a bijection $\phi : L \to L'$. By the preceding lemma, there exists some $z \notin L \cup L'$, so for any $x \in L$, we let $\phi(x)$ be the point in $L' \cap \overline{zx}$. The function ϕ is well-defined, since $\overline{zx}$ exists by (P2), and $|L' \cap \overline{zx}| = 1$ by (P1).

Now we prove that ϕ is a bijection. Let $y \in L'$ be given. It suffices to show that $|\phi^{-1}(y)| = 1$. Let $x \in L \cap \overline{yz}$. Then the set $\{x, z\} \subseteq \overline{yz} \cap \overline{xz}$, i.e. $|\overline{yz} \cap \overline{xz}| \geq 2$, so by (P1), $\overline{yz} = \overline{xz}$. This means that $\phi(x) = y$ and x is unique by (P1), so $|\phi^{-1}(y)| = 1$. ∎

The *order* of a finite projective plane $(X, \mathcal{L})$ is $|L| - 1$ for any $L \in X$.

Theorem. *Let $(X, \mathcal{L})$ be a finite projective plane of order n. Then the following statements all hold:*

1. *Exactly $n + 1$ lines pass through any point $x \in X$.*
2. *$|X| = n^2 + n + 1$.*
3. *$|\mathcal{L}| = n^2 + n + 1$.*

Proof. First we show that for any $x \in X$, there exists some line $L \in \mathcal{L}$ such that $x \notin L$. By (P0), there exists some $F = \{a, b, c, d\}$ and without loss of generality we will assume that $x \notin \{a, b, c\}$. So $\overline{ab} \cap F = \{a, b\}$ and $\overline{ac} \cap F = \{a, c\}$. Now if $x = d$, then $x \notin \overline{ab}$ and we're done. If $x \in \overline{ab}$, then $x \notin \overline{ac}$ and if $x \in \overline{ac}$, then $x \notin \overline{ab}$, since $|\overline{ab} \cap \overline{ac}| = 1$.

Let $x \in X$ be given and let L be a line in $\mathcal{L}$ such that $x \notin L$. Now we may prove each part of the theorem.

1. For all $y \in L$, there exists a line $\overline{xy}$ passing through x. There are $n + 1$ points $y \in L$, so there are at least $n+1$ lines passing through x. Now for all $L' \in \mathcal{L}$ such that $x \in L'$, $|L \cap L'| = 1$ by (P1), so L' was already counted above. So there are exactly $n + 1$ lines through x.
2. Let $L_i = \overline{xx_i}$ where $x_i \in L, i \in \{1, \ldots, n + 1\}$. By (P1), $L_i \cap L_j = \{x\}$ for all $i \neq j$. As $|L_i \setminus \{x\}| = n$ we have
$$|X| \geq |\bigcup_{i=1}^{n+1} L_i| = (n + 1)n + 1 = n^2 + n + 1,$$
so $|X| \geq n^2 + n + 1$. Now we show that for all $p \in X$, there exists an $i \in \{1, \ldots, n+1\}$ such that $p \in L_i$. If $p = x$, this is clearly true, so assume $p \neq x$. Then $\overline{px} \cap L = \{x_i\}$ for some i by (P1). So $\overline{px} = L_i$ for some i and we already counted p above. So $|X| = n^2 + n + 1$.
3. This follows from item 2 and duality, which we will introduce next. ∎

The *incidence graph* of a finite projective plane $(X, \mathcal{L})$, is a bipartite graph $G = (V, E)$ with $V = X \cup \mathcal{L}$ and $E = \{xL : \text{for all pairs } x, L \text{ such that } x \in L\}$. The concept of duality involves switching the roles of X and $\mathcal{L}$ in this bipartite graph, interpreting X as lines and $\mathcal{L}$ as points.

Formally, the *dual* of a finite projective plane $(X, \mathcal{L})$ is the set system $(\mathcal{L}, \Lambda)$ where $\Lambda \in 2^{\mathcal{L}}$ contains an element $\{L \in \mathcal{L} : x \in L\}$ for every $x \in X$.

Lemma. *The dual $(\mathcal{L}, \Lambda)$ of a finite projective plane $(X, \mathcal{L})$ is also a finite projective plane.*

Proof. We show that $(\mathcal{L}, \Lambda)$ satisfies each of (P0), (P1), and (P2).

P1) We need to find lines L_1, L_2, L_3, L_4 such that any for any $\lambda \in \Lambda$, with $\lambda = \{L \in \mathcal{L} : x \in L\}$ for some $x \in X$, the point x is contained in at most two of L_1, L_2, L_3, L_4. Let $F = \{a, b, c, d\}$ be the set given by applying (P0) to $(X, \mathcal{L})$. Let $L_1 = \overline{ab}, L_2 = \overline{cd}, L_3 = \overline{ad}, L_4 = \overline{bc}$. Suppose, towards a contradiction, that x is in three of these lines. Without loss of generality, suppose $x \in L_1 \cap L_2 \cap L_3$. By (P1) for $(X, \mathcal{L})$, $|L_i \cap L_j|$ for all $i \neq j$. Since $x \in L_1 \cap L_3$, we have that $x = a$. But since $x \in L_2 \cap L_3$, we have that $x = d$. This is a contradiction, since $a \neq d$. The same is true for any other triple of these four lines.

P2) Let $\lambda_1, \lambda_2 \in \Lambda, \lambda_1 \neq \lambda_2$ be given, i.e. $\lambda_1 = \{L \in \mathcal{L} : x_1 \in L\}$ and $\lambda_2 = \{L \in \mathcal{L} : x_2 \in L\}$ for some points $x_1 \neq x_2$. We need that $|\lambda_1 \cap \lambda_2| = 1$, but this follows from (P2) for $(X, \mathcal{L})$, since x_1 and x_2 intersect at exactly one line.

P3) Let $L_1, L_2 \in \mathcal{L}$ with $L_1 \neq L_2$. It suffices to show that there exists a unique $\lambda \in \Lambda$ such that $L_1 \in \lambda$ and $L_2 \in \lambda$. But since $\lambda = \{L \in \mathcal{L} : x \in L\}$ for some x, this is exactly (P1) for $(X, \mathcal{L})$. ∎

Corollary. *The order of the dual finite projective plane is the same as the order of the original.*

Proof. For any $\lambda \in \Lambda$ of the dual, $|\{L \in \mathcal{L} : x \in L\}| = n + 1$. This implies that the order or $(\mathcal{L}, \Lambda)$ is n. ∎

One may wonder whether there is a finite projective plane of any order. The answer is no. There are no finite projective planes of order 1, 6, or 10. There are finite projective planes of order 3, 4, 5, 7, 8, 9, and 11. Whether there is a finite projective plane of order 12 is an open problem. The existence of finite projective planes of some higer orders is known though, as stated by the following theorem.

Theorem. *A finite projective plane exists of order n if n is a prime power.*

We will not present the proof, which follows from $\mathbf{F}_n$ being a field. ∎

A corollary of this theorem is that there are infinitely many finite projective planes. Recall that a graph on n vertices without C_4 as its subgraph has at most $(n^{3/2} + n)/2$ edges. Then a corollary of the infinity of finite projective planes is given below.

Corollary. *For infinitely many values of n there is a graph on n vertices without C_4 as a subgraph that contains at least $(n/2)^{3/2}$ edges.*

Proof. Consider the incidence graph of the finite projective plane $(X, \mathcal{L})$. The number of vertices $n = |X| + |\mathcal{L}| = 2(n^2 + n + 1)$. Then for all $L \in \mathcal{L}$, since $|L| = m + 1$, the degree of L in G is $m + 1$ as well. This means that $|E| = (m + 1)|\mathcal{L}| = (m + 1)(m^2 + m + 1) \geq (m^2 + m + 1)^{3/2} = (n/2)^{3/2}$. So the incidence graph has at least $(n/2)^{3/2}$ edges, so we just need to show that it does not have C_4 as a subgraph. This is easy because a C_4 in the bipartite incidence graph implies that the intersection of two lines $L \cap L'$ contains two points x, x', which contradicts (P1). ∎

6. LATIN SQUARES

A *Latin square* of order n is a matrix $A \in \{1, \ldots, n\}^{n \times n}$ such that for any row r, $A_{ri} \neq A_{rj}$ if $i \neq j$ and for any column c, $A_{ic} \neq A_{jc}$ if $i \neq j$. Two Latin squares A and B are *orthogonal* if $(A_{ij}, B_{ij}) \neq (A_{xy}, B_{xy})$ whenever $i \neq x$ or $j \neq y$. Note that the number of ordered pairs of $\{1, \ldots, n\}$ is n^2 and there are only n^2 cells, so if all the pairs are different, each pair appears exactly once.

Theorem. *Let M be a set of pairwise orthogonal Latin squares of order n. Then $|M| \leq n - 1$.*

Proof. Let A, B be orthogonal Latin squares of order n and let π be some permutation of $\{1, \ldots, n\}$. Consider A' where $A'_{ij} = \pi(A_{ij})$. Note that $(A'_{ij}, B_{ij}) = (A'_{xy}, B_{xy})$ if and only if $(A'_{ij}, B_{ij}) = (A'_{xy}, B_{xy})$. So A' and B are also orthogonal.

So let $M = \{A_1, \ldots, A_t\}$ and for each $A_i \in M$, permute the n columns such that the first row of the resulting Latin square A'_i is $(1, 2, 3, \ldots, n)$. By our earlier observation, $A'_1, \ldots, A'_k$ are pairwise orthogonal. Now we zoom in on the second row, first column if A'_i for some i. Call this cell k. First we notice that $k \neq 1$, since the first element of the first row is 1. Then since comparing any two A'_i pairwise, we get all the pairs $(1, 1), (2, 2), \ldots, (n, n)$ in the first row, each of $2, 3, \ldots, n$ can appear in the second row, first column of at most one A'_i. So $t = |M| \leq n - 1$. ∎

The following theorem relates orthogonal Latin squares to finite projective planes.

Theorem. *For $n \geq 2$, a finite projective plane of order n exists if and only if there exists a set of $n - 1$ pairwise orthogonal Latin squares of order n.*

7. RAMSEY THEORY

Ramsey theory deals with the general question: "How big must a mathematical structure have to be such that some property holds?"

7.1. Independent sets and cliques

In a graph $G = (V, E)$, an *independent set* is a set $I \subseteq V$ such that $uv \notin E$ for all $u, v \in I$. A *clique* is a set $K \subseteq V$ such that $uv \in E$ for all $u, v \in K, u \neq v$. The *independent set number* $\alpha(G)$ is the maximum size of any independent set and the *clique number* $\omega(G)$ is the maximum size of any clique in G.

Theorem. *Let $G = (V, E)$ be a graph. If $|V| \geq \binom{k-l-2}{k-1} = \binom{k+l-2}{l-1}$, then $\alpha(G) \geq l$ or $\omega(G) \geq k$.*

Proof. By induction on $k + l$. In the base case $k = 1$ or $l = 1$, all the theorem says is that there is an independent set or clique of size 4. For the inductive step, let $k, l \geq 2$ and assume that the claim holds for $k, l-1$ or $k-1, l$. By Pascal's formula,

$$\binom{k+l-2}{k-1} = \binom{k+l-3}{k-1} + \binom{k+l-3}{k-2}.$$

Call the left-hand side n and from the right-hand side, call the first summand n_1 and the second summand n_2. Now pick any vertex $u \in V$. Let $B = N(u)$ and $A = V \setminus (B \cup \{u\})$. First note that it is impossible that both $|A| < n_1$ and $|B| < n_2$ since that would imply that

$$n = 1 + |A| + |B| \leq 1 + (n_1 - 1) + (n_2 - 1) = n - 1,$$

a contradiction. So either $|A| \geq n_1$ or $|B| \geq n_2$.

If $|A| \geq n_1$, by the induction hypothesis $\omega(G[A]) \geq k$ or $\alpha(G[A]) \geq l - 1$. Then a clique in $G[A]$ is also a clique in G, for the first case; and for the second case, adding the vertex u to an independent set in $G[A]$ gives an independent set in G, so $\alpha(G) \geq l$.

If it happens instead that $|B| \geq n_2$, then by induction we have either $\omega(G[B]) \geq k - 1$ or $\alpha(G) \geq l$. In the first of these cases, an independent set in $G[B]$ is also an independent set in G, so $\alpha(G) \geq l$; and in the second case, adding u to a clique in $G[B]$ gives a clique in G, so $\omega(G) \geq K$. ∎

For each $k, l \in \mathbf{N}$, the *Ramsey number* $r(k, l)$ is the minimum n such that for any graph G with n vertices, $\omega(G) \geq k$ or $\alpha(G) \geq l$.

Corollary.

$$r(k, l) \leq \binom{k+l-1}{k-1} = \binom{k+l-2}{l-1}.$$

Very little is known about Ramsey numbers. By observation, we have that $r(k, 1) = r(1, l) = 1$; $r(2, l) = l$ and $r(k, 2) = k$. For $k = l$, we only know that $r(3, 3) = 6$ and $r(4, 4) = 18$. The exact values of $r(n, n)$ for $n \geq 5$ are unknown. The above corollary gave an upper bound and the next theorem gives a lower bound.

Theorem. *For $k \geq 3$, $r(k, k) > 2^{k/2}$.*

Proof. First we introduce some basic notions from probability that we will need. Let Ω be a probability space and let X be an event. Then $\Pr[X] = |X|/|\Omega| < 1$ if and only if $|X| < |\Omega|$, which would imply that $\neg X \neq \emptyset$.

Now we create a random graph G by starting with a set of n vertices and adding each edge with probablity $1/2$ uniformly at random. We're trying to prove that if $n \leq 2^{k/2}$ there is a G with $|V| = n$ such that $\alpha(G) < k$ and $\omega(G) < k$, so it suffices to show that

$$\Pr[\alpha(G) \geq k \text{ or } \omega(G) \geq k : n \leq k^{k/2}] < 1.$$

Let A be a subset of vertices with $|A| = k$. Let K_A denote the proposition "A forms a clique" and let I_A denote the proposition "A forms an independent set". Then let $X_A = K_A \cup I_A$. $\Pr[K_A] = (1/2)^{\binom{k}{2}} = \Pr[I_A]$ and since $K_A \cap I_A = \emptyset$,

$$X_A = 2 \cdot \left(\frac{1}{2}\right)^{\binom{k}{2}} = 2^{1-\binom{k}{2}}.$$

Let $\Pr(Y)$ denote the probability that $\alpha(G) \geq k$ or $\omega(G) \geq k$. This is equal to the probablilty that X_A holds for some $A \subseteq V$ with $|A| \geq k$. So

$$\Pr[Y] \leq \sum_{\substack{A \subseteq V \\ |A| \geq k}} 2^{1-\binom{k}{2}} = \binom{n}{2} 2^{1-\binom{k}{2}} \leq \frac{n^k}{k!} 2^{1-\binom{k}{2}},$$

and we can further derive the strict inequality

$$\frac{n^k}{k!}2^{1-\binom{k}{2}} < \frac{n^k}{2^{k/2+1}}2^{1-(k^2/2-k/2)} = \left(\frac{n}{2^{k/2}}\right)^k,$$

which is less than 1, since $n \leq 2^{k/2}$. ∎

7.2. Increasing/decreasing subsequences

Given a finite sequence $S = (x_1, x_2, \ldots, x_n)$ of numbers, an *increasing* (*decreasing*) *subsequence* of length k is a sequence $(x_{i_1}, x_{i_2}, \ldots, x_{i_n})$ such that for all j, l with $1 \leq j < l \leq k$, $i_j < i_l$ and $x_{i_j} \leq x_{i_l}$ ($x_{i_j} \geq x_{i_l}$).

Theorem E (*Erdős-Szekeres*). *For any finite sequence S of at least $(r+1)(s+1)+1$ numbers, there is an increasing subsequence of length r or a decreasing subsequence of length s.*

Proof. Suppose, towards a contradiction, that every increasing/decreasing subsequence has length at most $r-1/s-1$ respectively. Then we label each x_i of S with a pair (a_i, b_i) where a_i/b_i is the length of the longest increasing/decreasing subsequence ending in x_i.

We claim that each of the numbers x_i, x_j have different labels. Let $i \neq j$ and without loss of generality, assume $i < j$. If $x_i \leq x_j$, then $a_i < a_j$, as any increasing subsequence ending in x_i can be extended by x_j. Likewise, if $x_i \geq x_j$, then $b_j > b_i$, as any decreasing subsequence ending in x_i can be extended by x_j.

By our assumption, $a_i \leq r-1$ and $b_i \leq s-1$ for all i. So the number of labels $(a_i, b_i) \leq (r-1)(s-1)$ which implies that the length of the sequence S is at most $(r-1)(s-1)$, a contradiction. ∎

The given bound is tight; there are sequences with length $(r-1)(s-1)$ numbers that neither have an increasing subsequence of length r, nor have a decreasing subsequence of length s. To construct one, take a grid of $(r-1)(s-1)$ points in the plane and rotate it *very* slightly counterclockwise such that no two points are on the same horizontal or vertical line. Then we have a set of points $(x_i, y_i), \ldots (x_n, y_n)$ where $n = (r-1)(s-1)$. Now order the y-coordinates *by the value of the x-coordinates*. Any increasing subsequence can take at most one element from each column, so at most $r-1$, and any decreasing subsequence can take at most one element from each row, so at most $s-1$.

8. ERROR-CORRECTING CODES

Suppose we have two people who need to communicate over a channel that is "noisy" in the sense that errors may be introduced into the message. To detect and solve this issue, checksums may be used. Suppose we have an alphabet $\Sigma = \{0, \ldots, q-1\}$ where q is prime. Let x denote a message $x \in \Sigma^k$ for some positive integer k. The *checksum* c of x is $\sum_{i=1}^{k} x_i \pmod q$ and we can encode te message as $y \in \mathbf{F}_q^{k+1}$ where

$$y_i = \begin{cases} x_i & \text{for } i \leq k \\ c & \text{for } i = k+1 \end{cases}$$

Since q is prime, for any $x, y \in \mathbf{F}_q^k k$, if x and y differ at *exactly one* place, then their checksums will differ. But what happens if more errors occur? Also, is it possible for the receiver to determine where the error occured without asking the sender to resend the message?

To formalise the problem further, we introduce some more definitions. The *block length* n is the length of the encoded message. Suppose the original (non-encoded) message has length k. Then the *encoding function* $E : \Sigma^k \to \Sigma^n$ is applied by the sender before transmission and the *decoding function* $D : \Sigma^n \to \Sigma^k$ is applied by the receiver upon the message's arrival. The *code* $C \subseteq \Sigma^n$ is the image of E. A code is *binary* if $\Sigma = \{0, 1\}$. Checksums work by increasing the distance between two messages $x, y \in \Sigma^k$ if $x \neq y$.

The *Hamming distance* between two messages $x, y \in \Sigma^m$ is given by

$$\Delta(x, y) = \sum_{i=1}^{m} [x_i \neq y_i].$$

For a code $C \subseteq \Sigma^n$, the *minimum distance* of C is

$$\Delta(C) = \min_{x,y \in C} \{\Delta(x,y)\}.$$

An $(n,k,d)_q$*-code* is a code C such that

i) $C \subseteq \Sigma^n$;
ii) $k = \log_q |C|$;
iii) and $d = \Delta(C)$.

Note that $|C|$ is the size of the code and k need not be an integer. A checksum is a $(n, n-1, 2)_q$-code: Suppose that $x \in C$. Let y be such that $\Delta(x,y) = 1$. Is it possible that $y \in C$? No, because for all $x, y \in C$, if $x \neq y$ then $\Delta(x,y) \geq 2$.

If, for a code C, we can be sure that at most r errors occur during transmission, then:

i) If $\Delta(C) \geq r+1$, then the error can be *detected* because for any $x \in C$, if we have y that differs from x at not less than $r+1$ places, i.e. $\Delta(x,y) \leq r$, then $y \notin C$.
ii) If $r \leq \lfloor (\Delta(C)-1)/2 \rfloor$, then the error can be *corrected*. Construct a graph with $V = \Sigma^n$ with an edge between $x, y \in \Sigma^n$ if $\Delta(x,y) = 1$. Now if we have $y \in \Sigma^n$ which we know is a garbled message, we can find the closest valid message in the graph. If the error is not too large, then the closest valid message will have been the intended one.

The general aim is to construct codes with small n and large $\Delta(C)$. For any finite projective plane $(X, \mathcal{L})$ of order m, we can construct a $(m^2+m+1, \log_2(m^2+m+1), 2m)_2$-code by associating every letter in the alphabet with a characteristic vector of $L \in \mathcal{L}$. For example, the letter associated with the line $\{1,5,6\}$ in the Fano plane will be coded as 1000110. Since for any $L_1, L_2 \in \mathcal{L}, L_1 \neq L_2$, $|L_1 \cap L_2| = 1$ and each of L_1, L_2 have $m+1$ points, $\Delta(y_{L_1}, y_{L_2}) = 2m$, where y_{L_i} is the letter associated with the line L_i.

A *linear* code $C \subseteq \mathbf{F}_q^n$ is a linear subspace of $\mathbf{F}_q$, i.e. for all $x, y \in C$, $\alpha \in \mathbf{F}_q$, $\alpha x + y \in C$. There exist k independent vectors $x_1, \ldots, x_k \in \mathbf{F}_q^n$ such that

$$C = \left\{ \sum_{i=1}^{k} \alpha_i x_i : \alpha_i, \ldots \alpha_n \in \mathbf{F}_q \right\}.$$

More succinctly, we can express C using the generator matrix $G \in \mathbf{F}_q^{n \times k}$. The matrix $G = \begin{matrix} x_1 & \cdots & x_k \end{matrix}$ and $C = \{G_\alpha : \alpha \in \mathbf{F}_q^k\}$. (Multiplying any message by G on the left encodes it.)

Alternatively, we may represent C as the *null space* of a matrix. Recall that for any k-dimensional linear subspace C, there is an $(n-k)$-dimensional linear subspace $C^\perp$ such that for all $x \in C, y \in C^\perp$, $x^T y = 0$. Using the generator matrix $H \in \mathbf{F}_k^{n \times (n-k)}$ of $C^\perp$, express $C = \{x : x^T H = 0\}$. Then H is called the *parity check* matrix of C. In a sense, linear codes are simple because we can detect errors simply by computing $x^T H$.

The *support* of a vector x is the set $\{i : x_i \neq 0\}$ and the *Hamming weight* of x is $\operatorname{wt} x = |\{i : x_i \neq 0\}|$.

Theorem. *For a linear code C,*

$$\Delta(C) = \min\{\operatorname{wt} x : x \in C \text{ and } x \neq 0\}.$$

Proof. For ease of notation, let d denote $\min\{\operatorname{wt} x : x \in C \text{ and } x \neq 0\}$. First we show that $d \leq \Delta(C)$. Let $y, z \in C$ be such that $\Delta(y,z) = \Delta(C)$. Since C is linear, $x = y - z \in C$. Note that $\Delta(y,z) = |\{i : x_i \neq 0\}|$, so $d \leq \operatorname{wt} x = \Delta(C)$.

Now we show that $d \geq \Delta(C)$. Note that $0 \in C$. Let $x \neq 0$ be such that $\operatorname{wt} x = d$. Then $\Delta(C) \leq \Delta(0,x) = |\{i : x_i \neq 0\}| = \operatorname{wt} x = d$. ∎

We can use this theorem to construct a linear code with $\Delta(C) \geq 3$. We find a parity check matrix H such that $x^T H \neq 0$ for any non-zero x with $\operatorname{wt} x \in \{1,2\}$. For simplicity, we will work with $q = 2$. If $\operatorname{wt} x = 1$, then X has exactly one 1-entry, say it is at position i. So $x^T H$ is the i-th row H_i of H. We need

that H_i is non-zero for every i. If $\operatorname{wt} x = 2$, then x^T has exactly two non-zero elements, say they are at positions i and j. Then $x^T H = (H_i + H_j)^T$ hence we need $H_i \neq H_j$ for all $i \neq j$.

If all rows of H are distinct and non-zero, then we get $\Delta(C) \geq 3$. The largest number of such rows if $H \in \mathbf{F}_2^{n \times l}$ is $2^l - 1$ so $n = 2^l - 1$ for any $l = n - k$. Hence we obtain an $(n, n - \log_2(n+1), 3)_2$-code called the *Hamming code*.

Let $x \in \mathbf{F}_q^n$. The *ball* around x of radius r is given by

$$B(x, r) = \{y \in \mathbf{F}_q^n : \Delta(x, y) \leq r\}.$$

The *volume* of such a ball is

$$\operatorname{vol}(r, n) = |B(x, r)| = \sum_{i=0}^{r} \binom{n}{i} (q-1)^i.$$

Theorem S (*Hamming bound*). *If an $(n, k, d)_q$-code exists, then*

$$q^k \cdot \operatorname{vol}\left(\left\lfloor \frac{d-1}{2} \right\rfloor, n\right) \leq q^n.$$

Proof. Let $r = \lfloor (d-1)/2 \rfloor$ and consider $\bigcup_{x \in C} B(x, r) \subset \{1, \ldots, q\}^n$. By observation,

$$\left| \bigcup_{x \in C} B(x, r) \right| = \sum_{x \in C} \operatorname{vol}(r, n) = q^k \cdot \operatorname{vol}(r, n) \leq q^n. \quad \blacksquare$$

Note that if $d = 3$ and $q = 2$ then the Hamming bound is $2^k(n+1) \leq 2^n$ and the Hamming code matches this bound with equality as $k = n - log_2(n+1)$. An $(n, k, d)_2$-code is called *perfect* if $q^k \cdot \operatorname{vol}(\lfloor (d-1)/2 \rfloor, n) = q^n$. There are no codes with larger size k than a $(n, n-d+1, d)_2$-code (even if we let $q > 2$), as the following theorem shows.

Theorem B (*Singleton bound*). *If C is an $(n, k, d)_q$-code, then $k \leq n - d + 1$.*

Proof. Define a function $f : \Sigma^n \to \Sigma^{k-1}$ such that $f(x_1, \ldots, x_n) = (x_1, \ldots, x_{k-1})$. As $|\Sigma^{k-1}| = q^{k-1}$ and $|C| = q^k > q^{k-1}$, there are $x, y \in C$, with $x \neq y$ and $f(x) = f(y)$. So x and y can only differ in the last $n - k + 1$ entries. Since $d = \Delta(C) \leq \Delta(x, y) \leq n - k + 1$, $k \leq n - d + 1$. $\blacksquare$

An $(n, k, d)_q$ code is called *maximum-distance separable* or MDS if $k = n - d + 1$.