

UiO : Department of Mathematics
University of Oslo

MAT3250 Spring 2025

Discrete mathematics revision

May 23, 2025

Colour coding of slide titles

For the exam:

- **Green = Important to know**
- **Orange = Good to know (but I will avoid asking tricky questions using this material)**
- **Red = This was fun, but won't appear at the exam**

Counting

Enumeration

Main problem

Given a finite set S, ask for the number of elements $|S|$, or more formally the **cardinality** of S.

Example

How many words of lenght 3 using letters a, b, c exist, assuming no letter is immediately repeated in the word?

$$S = \{aba, abc, aca, acb, bab, bac, bca, bcb, cab, cac, cba, cbc\}, |S| = 12$$

Basic tools

- If there is a bijection between finite sets $A \leftrightarrow B$, then $|A| = |B|$.
- If a set A can be partitioned into disjoint subsets $A_1, \ldots, A_n$, then $|A| = \sum_{i=1}^{n} |A_i|$.
- Given sets A, B, then $|A \times B| = |A| \times |B|$.

Combining these

In order to count $\{a, b, c\}$-words of lenght three with no repetitions, can say: The first letter is chosen freely among the three, there are then two valid choices for the second letter, and then two valid choices for the third letter. This gives $3 \times 2 \times 2 = 12$ words.

Permutations

Permutation

Let S be a finite set. A bijection $\pi : S \to S$ is called a **permutation** of S.

Cycle

A **cycle** is a permutation π such that for some distinct elements $a_1, a_2, \ldots, a_k \in S$,

$$\pi(a_1) = a_2, \quad \pi(a_2) = a_3, \quad \ldots, \quad \pi(a_{k-1}) = a_k, \quad \pi(a_k) = a_1,$$

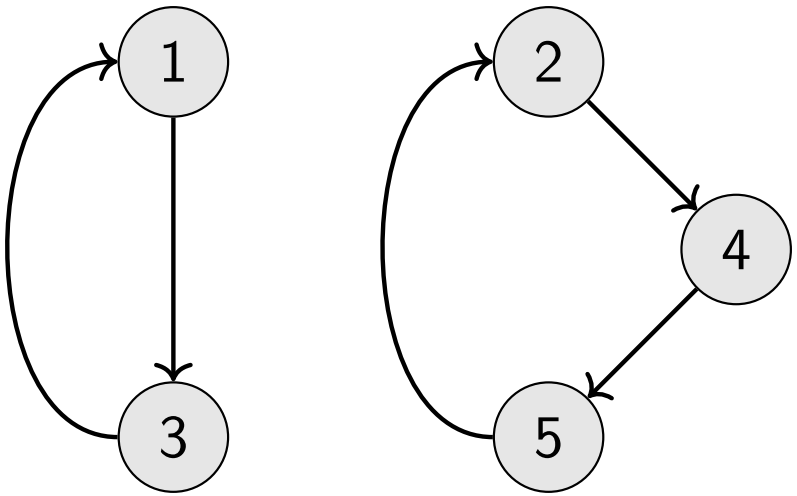
and $\pi(x) = x$ for all $x \notin \{a_1, \ldots, a_k\}$. We write in "cycle notation" $\pi = (a_1 a_2 \ldots a_k)$.

Cycle decomposition

Disjoint Cycles

Every permutation of a finite set can be written as a product of **disjoint cycles**, i.e. cycles involving only different elements.
Example: If $\pi(1) = 3, \pi(2) = 4, \pi(3) = 1, \pi(4) = 5, \pi(5) = 2$, then:

$$\pi = (1\,3)(2\,4\,5).$$



Counting numbers

A set of size m is called an m-set.

Binomial coefficients

The number of k-subsets of an n-set is the **binomial coefficient**

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Rising and falling factorials

Rising and falling factorials

For $x \in \mathbb{R}$ and $m \geq 0$, have the **rising factorial**

$$x^{\overline{m}} = x(x+1)\dots(x+m-1)$$

and the **falling factorial**

$$x^{\underline{m}} = x(x-1)\dots(x-m+1).$$

Example

We have

$$\binom{n}{k} = \frac{n^{\underline{k}}}{k!}.$$

Stirling numbers

Stirling numbers of the first kind

The number of permutations π of $\{1, \ldots, n\}$ which decompose into precisely k cycles is the **Stirling number of the first kind**, denoted $s_{n,k}$.

Stirling numbers of the second kind

The number of ways to partition $\{1, \ldots, n\}$ into k nonempty subsets is the **Stirling number of the second kind**, denoted $S_{n,k}$.

Stirling Numbers examples

Stirling Numbers of the First Kind $s_{n,k}$

- $s_{3,1} = 2$ (the 3-cycles: $(123), (132)$)
- $s_{3,2} = 3$ (permutations like $(1)(23), (2)(13), (3)(12)$)
- $s_{3,3} = 1$ (the identity permutation $(1)(2)(3)$)

Stirling Numbers of the Second Kind $S_{n,k}$

- $S_{3,1} = 1$ (the full set: $\{1, 2, 3\}$)
- $S_{3,2} = 3$ (e.g., $\{1, 2\}, \{3\}$, etc.)
- $S_{3,3} = 1$ (singleton sets: $\{1\}, \{2\}, \{3\}$)

Generating functions

Definition

Given a sequence of numbers $(a_n)_{n=0}^{\infty}$, its **generating function** is the power series $A(z) = \sum_{n=0}^{\infty} a_n z^n$.

Operations on generating functions

Let $(a_n), (b_n)$ be sequences with generating functions $A(z), B(z)$.

- The generating function of $(a_n + b_n)$ is $A(z) + B(z)$
- The generating function of $(\sum_{i=0}^{n} a_i b_{n-i})_{n=0}^{\infty}$ is $A(z)B(z)$
- The generating function of the sequence

$$c_0 = 0, c_i = a_{i-1} \text{ if } i \geq 1$$

is $zA(z)$.

Recurrence Relations

Definition

A **recurrence relation** is an equation that expresses each term of a sequence as a function of its preceding terms.

Examples

- **Fibonacci sequence:**

$$F_0 = 0, \quad F_1 = 1, \quad F_n = F_{n-1} + F_{n-2} \quad \text{for } n \geq 2.$$

- **Factorials ($a_n = n!$):**

$$a_0 = 1, \quad a_n = n \cdot a_{n-1} \quad \text{for } n \geq 1.$$

Generating Functions and Recurrence

Theorem

Let $(a_n)_{n=0}^{\infty}$ be a sequence satisfying a recurrence relation of the form

$$a_n + c_1 a_{n-1} + \cdots + c_k a_{n-k} = 0 \quad \text{for } n \geq k,$$

with c_i numbers. Then the generating function $A(z) = \sum_{n=0}^{\infty} a_n z^n$ can be written as

$$A(z) = \frac{P(z)}{Q(z)},$$

where $Q(z) = z^k + c_1 z^{k-1} + \cdots + c_k$ and $P(z)$ is a polynomial of degree less than k.

Partial fraction decompositions

Taking the partial fraction decomposition of $A(z) = P(z)/Q(z)$ gives a formula for a_n. Let $\alpha_1, \ldots, \alpha_l$ be the roots of $Q(z^{-1})$, and let e_i be the multiplicity of α_i. Then we can find $b_{i,e}$ such that

$$A(z) = \frac{P(z)}{Q(z)} = \sum_{i=1}^{l} \sum_{e=1}^{e_i} \frac{b_{i,e}}{(1-\alpha_i z_i)^e}$$

gives

$$a_n = \sum_{i=1}^{k} b_{i,e} \binom{e+n-1}{e-1} \alpha_i^n$$

Example

Example

The Fibonacci sequence satisfies $F_n = F_{n-1} + F_{n-2}$. Its generating function is

$$F(z) = \frac{z}{1 - z - z^2} = \frac{1/\sqrt{5}}{1 - \alpha_1 z} - \frac{1/\sqrt{5}}{1 - \alpha_2 z},$$

where $\alpha_1 = \frac{1+\sqrt{5}}{2}$ and $\alpha_2 = \frac{1-\sqrt{5}}{2}$. This gives

$$F_n = \frac{1}{\sqrt{5}}\alpha_1^n - \frac{1}{\sqrt{5}}\alpha_2^n$$

Exponential generating functions

Definition

Let $(a_n)_{n=0}^{\infty}$ be a sequence. The **exponential generating function** of this sequence is the power series

$$G(z) = \sum_{n=0}^{\infty} \frac{a_n}{n!} z^n.$$

Example

The exponential generating function for the constant sequence $a_n = 1$ is

$$G(z) = \sum_{n=0}^{\infty} \frac{1}{n!} z^n = e^z.$$

The inclusion-exclusion principle

Inclusion-Exclusion Principle

For finite sets $A_1, A_2, \ldots, A_n$, the size of their union is:

$$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_i |A_i| - \sum_{i<j} |A_i \cap A_j| + \sum_{i<j<k} |A_i \cap A_j \cap A_k| - \cdots + (-1)^{n+1} |A_1 \cap A_2 \cap \cdots \cap A_n|.$$

Examples

Two sets: $|A \cup B| = |A| + |B| - |A \cap B|$.
Three sets:

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|.$$

Asymptotic Analysis and Big O Notation

Big O Notation

Let $f(n)$ and $g(n)$ be functions from $\mathbb{N}$ to $\mathbb{R}_{\geq 0}$. We write:

$f(n) = O(g(n))$ if there exist constants $C > 0$ and $n_0 \in \mathbb{N}$ such that

$$f(n) \leq C \cdot g(n) \quad \text{for all } n \geq n_0.$$

Examples

- $5n^2 + 3n + 7 = O(n^2)$
- $\log n = O(n)$, but $n \neq O(\log n)$
- If an algorithm takes $\leq c \cdot n \log n$ steps, its running time is $O(n \log n)$

Graph theory

Graphs

Definition

A **graph** G consists of:

- a set of **vertices** V, and
- a set of **edges** $E \subseteq \binom{V}{2} = \{\{u, v\} \mid u, v \in V, u \neq v\}$.

We write $G = (V, E)$.

Example

Let $V = \{1, 2, 3\}$ and $E = \{\{1, 2\}, \{2, 3\}\}$. This graph has 3 vertices and 2 edges.



Basic Graph Terminology & notation

Incidence and Adjacency

- An edge $e \in E$ is **incident** to a vertex v if $v \in e$.
- Often write uv for the edge $\{u, v\}$.
- Two vertices u and v are **adjacent** if $uv \in E$.

Order and Size of a Graph

Let $G = (V, E)$ be a graph.

- The **order** of the graph is $n = |V|$, the number of vertices.
- The **size** of the graph is $q = |E|$, the number of edges.

Degrees

Degree of a Vertex

The **degree** of a vertex v, denoted $\deg(v)$, is the number of edges incident to v.

Handshake Lemma

In any graph $G = (V, E)$, we have

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

More on degrees

Degree Sequence

The **degree sequence** of a graph is the list of vertex degrees written in non-increasing order:

$$d_1 \geq d_2 \geq \cdots \geq d_n,$$

where $V = \{v_1, \ldots, v_n\}$ and $d_i = \deg(v_i)$.

k-Regular Graph

A graph is called k-**regular** if every vertex has degree k.

Basic Examples of Graphs

Complete Graph K_n

A **complete graph** on n vertices, denoted K_n is a graph where every pair of distinct vertices is connected by an edge.

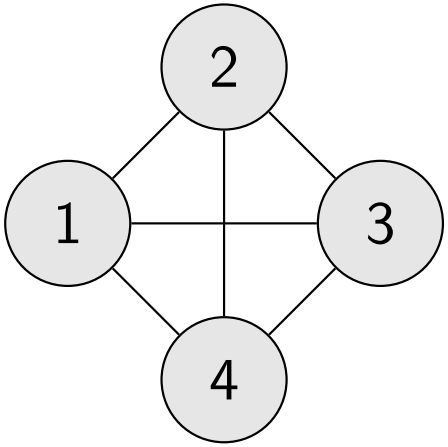
Path Graph P_n

A **path graph** on n vertices consists of a single path: a sequence of n vertices connected by $n-1$ edges in a straight line.

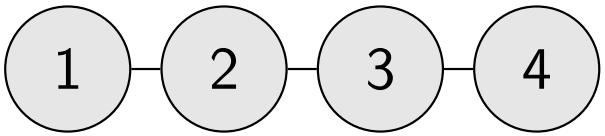
Circuit Graph C_n

A **circuit graph** (or **cycle graph**) on $n \geq 3$ vertices is a path where the two endpoints are also connected by an edge, forming a closed loop.

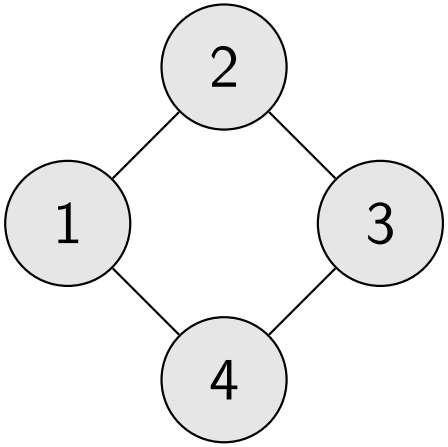
Complete graph K_4



Path graph P_4



Circuit graph C_4



Adjacency Matrix of a Graph

Definition

Let $G = (V, E)$ be a graph with n vertices. The **adjacency matrix** A of G is the $n \times n$ matrix where:

$$A_{ij} = \begin{cases} 1 & \text{if there is an edge between } v_i \text{ and } v_j, \\ 0 & \text{otherwise.} \end{cases}$$

Example

For the graph with edges $\{1, 2\}, \{2, 3\}, \{3, 1\}$, the adjacency matrix is:

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

Incidence Matrix of a Graph

Definition

Let $G = (V, E)$ be a graph with n vertices and q edges. The **incidence matrix** M is the $n \times q$ matrix where:

$$M_{ij} = \begin{cases} 1 & \text{if vertex } v_i \text{ is incident to edge } e_j, \\ 0 & \text{otherwise.} \end{cases}$$

Example

For the graph with $v = \{1, 2, 3\}$ and edges $e_1 = \{1, 2\},\ e_2 = \{2, 3\}$:

$$M = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}$$

Subgraphs and Induced Subgraphs

Subgraph

A graph $H = (V_H, E_H)$ is a **subgraph** of a graph $G = (V, E)$ if:

$$V_H \subseteq V \quad \text{and} \quad E_H \subseteq E.$$

That is, H is formed by removing vertices and/or edges from G.

Induced Subgraph

A subgraph $H = (V_H, E_H)$ is an **induced subgraph** if $E_H = E \cap \binom{V_H}{2}$, that is the edges of H are exactly the edges of G with endpoints in V_H. It is then the "largest" subgraph of G with vertex set V_H.

Paths and Circuits

Path

A **path** or **walk** of length k in a graph is a sequence of vertices

$$v_0, v_1, \ldots, v_k$$

such that each consecutive pair $\{v_{i-1}, v_i\} \in E$ is an edge of the graph.

Circuit

A **circuit** (or **cycle**) is a path in which the first and last vertices are equal:

$$v_0 = v_k, \quad \text{and all other } v_i \text{ are distinct.}$$

Connectedness

Definition

A graph $G = (V, E)$ is called **connected** if, for every pair of vertices $u, v \in V$, there exists a path in G from u to v.

Disconnected Graph

A graph that is not connected is called **disconnected**. It consists of two or more separate connected subgraphs (**components** of the graph) with no edges between vertices in different components.

Distance in a Graph

Definition

Let $G = (V, E)$ be a graph. The **distance** between two vertices $u, v \in V$, denoted $\mathrm{d}(u, v)$, is the length of the shortest path from u to v in G.

- If $u = v$, then $\mathrm{d}(u, v) = 0$.
- If u and v are not connected (i.e. lie in different components), then $\mathrm{d}(u, v) = \infty$.

Example

In the path graph $1 - 2 - 3 - 4$, we have:

$$\mathrm{d}(1, 4) = 3, \quad \mathrm{d}(2, 3) = 1.$$

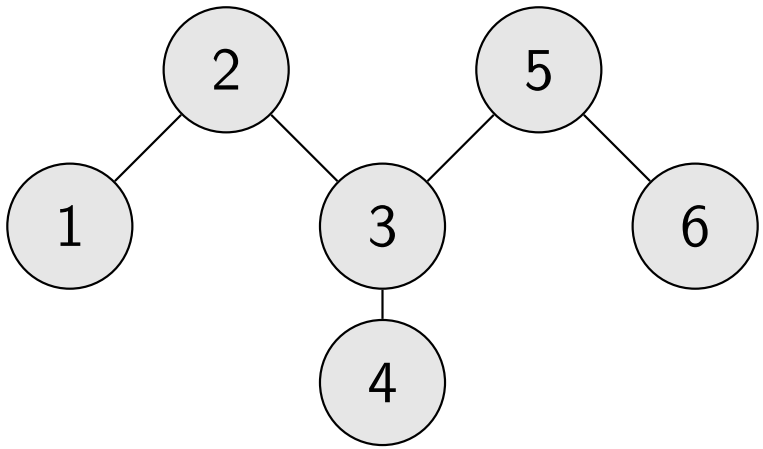
Trees

Definition

A **tree** is a connected graph with no cycles. A **forest** (or **acyclic graph**) is one where every connected component is a tree.

Theorem

A graph $G = (V, E)$ is a tree if and only if it is connected and $|E| = |V| - 1$.



Spanning Trees

Definition

A **spanning tree** of a connected graph $G = (V, E)$ is a subgraph $T \subseteq G$ such that:

- T is a **tree**, and
- T contains **all the vertices** of G.

Existence Theorem

Every connected graph has at least one spanning tree.

Cayley's formula and Prüfer sequences

The number of spanning trees in K_n is n^{n-2}. There is an explicit bijection

$$\{\text{Spanning trees in } K_n\} \leftrightarrow \{1, \ldots, n\}^{n-2},$$

mapping a tree T to its **Prüfer sequence**:

1. Start with the spanning tree T.
2. Given a tree in K_n, find the leaf vertex of with lowest vertex label.
3. Add the vertex label of the unique neighbour of v in the tree in the Prüfer sequence.
4. Delete v from the tree, and repeat from step 2.
5. Stop when the tree has 2 vertices.

Constructing a Spanning Tree via depth first search

Algorithm Sketch

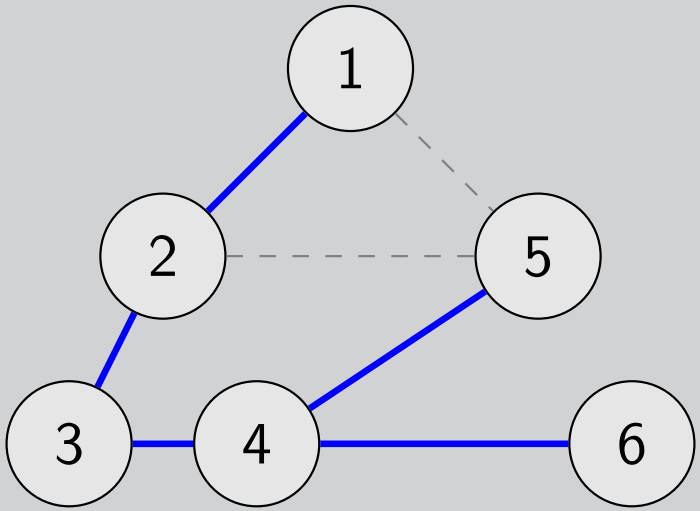
Build T inductively:

1. Start at any vertex v, add v to T and let v be the "current vertex".
2.
   - If the current vertex c has an unvisited neighbour w in G: Add w and edge cw to T. Record c as the parent of w. Update current vertex to w and repeat step 2.
   - If the current vertex c has no unvisited neighbour, "backtrack": Update current vertex to the parent of c. Repeat step 2.

The algorithm stops when $|V_T| = |V_G|$.

DFS Traversal and Spanning Tree Example

Graph with DFS Tree Starting from Vertex 1



Breadth-First Search (BFS)

Algorithm Sketch

Build T inductively:

1. Start from any vertex v, add it to T, let $v_1 = v$. Set the current vertex to v.
2. Assume $V_T = \{v_1, \ldots, v_i\}$ and current vertex is v_j. Say v_j has a neighbours $w_1, \ldots, w_a$ in G not already added to T. For $k = 1, \ldots, a$, let $v_{i+k} = w_k$, add v_{i+k} and $v_j v_{i+k}$ to T. Update current vertex to v_{j+1}, and repeat step 2.

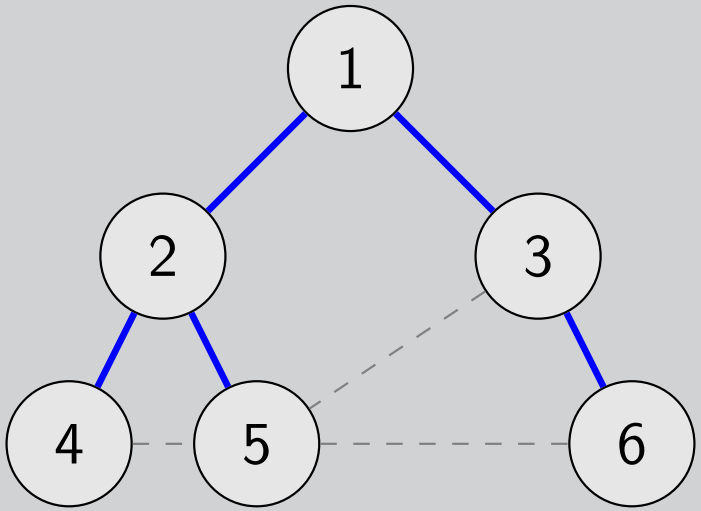
The algorithm stops when $|V_T| = |V_G|$.

Application

Starting from vertex v, moving in the breadth first spanning tree T gives a minimal length path from v to any other vertex.

BFS Traversal and Spanning Tree Example

Graph with BFS Tree Starting from Vertex 1



Kruskal's (greedy) algorithm

Setting

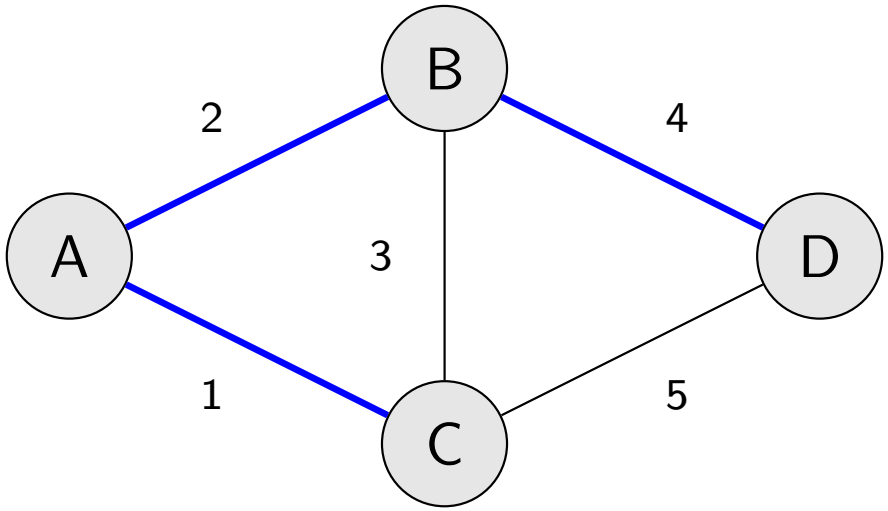
Let $G = (V, E)$ be a connected graph, and let $w\colon E \to \mathbb{R}_{\geq 0}$ be a weight function. Kruskal's algorithm finds a spanning tree T of minimal total weight

$$w(T) = \sum_{e \in E_T} w(e)$$

Algorithm

1. Start with $V_T = V$, and $E_T = \varnothing$.
2. Let $e_1, \ldots, e_k \in E$ be the edges which can be added to T without creating a cycle. Among these, add the one with minimal weight to E_T. Repeat step 2 with the new T.
3. Stop when T contains $|V| - 1$ edges.

Kruskal's Algorithm: Example



Construction

Kruskal's algorithm adds edges in increasing weight:

$$AC(1), \quad AB(2), \quad BD(4)$$

It skips $BC(3)$ since it would form a cycle.

Matroids

Definition (Matroid via Independent Sets)

A **matroid** is a pair $M = (S, \mathcal{U})$, where:

- S is a finite set (called the *ground set*),
- $\mathcal{U} \subseteq 2^S$ is a collection of subsets called the **independent sets**,

satisfying:

(1) $\emptyset \in \mathcal{U}$,

(2) If $A \in \mathcal{U}$ and $B \subseteq A$, then $B \in \mathcal{U}$,

(3) If $A, B \in \mathcal{U}$ and $|A| < |B|$, then there exists $e \in B \setminus A$ such that $A \cup \{e\} \in \mathcal{U}$ (exchange property).

Matroids II

Terminology

Let $\mathcal{M} = (S, \mathcal{U})$ be a matroid.

- A maximal set $B \in \mathcal{U}$ is called a **basis**.
- If B_1, B_2 are bases, then $|B_1| = |B_2| \stackrel{def}{=} r(\mathcal{M})$, the rank of $\mathcal{M}$.

Examples

The following are matroids:

- V a vector space, $S = \{v_1, \ldots, v_m\} \subseteq V$, $A \subseteq V$ is independent if the elements of A are linearly independent.
- $G = (V, E)$ a graph, $S = E$, and $A \subseteq E$ is independent if the graph (V, A) is acyclic (i.e. a forest).

Greedy Algorithm for Matroids

Greedy Algorithm

Given a matroid $M = (S, \mathcal{U})$ and a weight function $w\colon S \to \mathbb{R}$, the following algorithm finds a minimum-weight independent set.

Algorithm:

1. Sort elements of S in ascending order of weight

$$w(e_1) \leq w(e_2) \leq \cdots \leq w(e_n).$$

2. Start with the empty set $A = \emptyset$, set $i = 1$.
3. Add e_i to A if $A \cup \{e_i\} \in \mathcal{U}$.
4. Update i to $i + 1$, and repeat step 3.

Bipartite Graphs

Definition

A graph $G = (V, E)$ is **bipartite** if its vertex set can be partitioned into two disjoint sets S and T such that every edge connects a vertex in S to a vertex in T, and there are no edges within S or within T.

The complete bipartite graph

The bipartite graph with $|S| = m$, $|T| = n$ and an edge connecting every vertex in S to every vertex in T is the complete bipartite graph $K_{m,n}$.

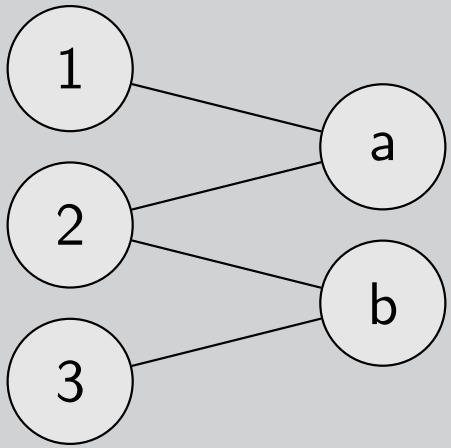
Theorem

A graph is bipartite if and only if every cycle has even length.

Bipartite graph example

Example

A bipartite graph with parts $S = \{1, 2, 3\}$ and $T = \{a, b\}$:



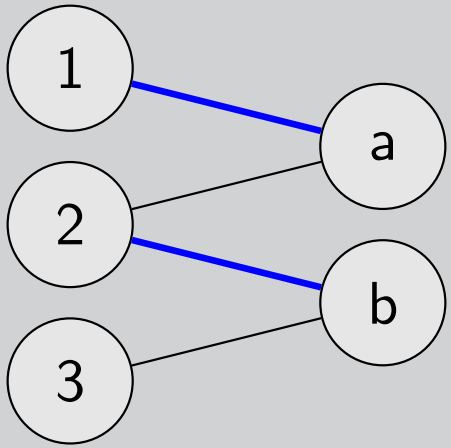
Matchings in Bipartite Graphs

Definition

A **matching** in a graph is a set of edges with no shared vertices.

In a bipartite graph $G = (S \cup T, E)$, a matching pairs vertices of S to distinct vertices of T.

Example



Hall's Marriage Theorem

Theorem

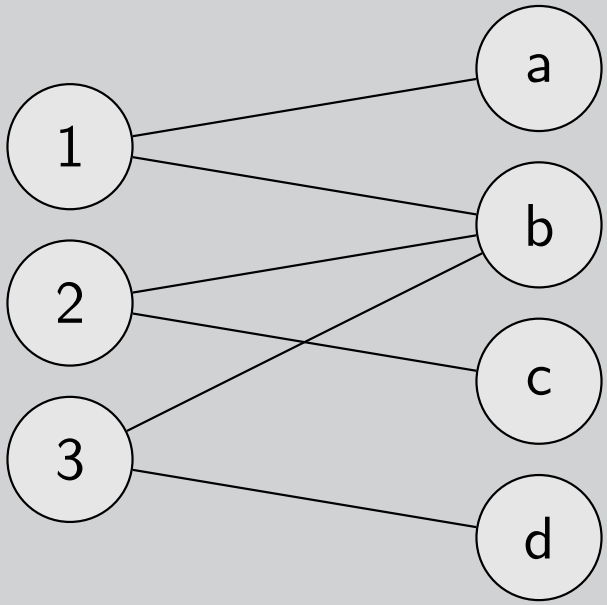
Let $G = (S \cup T, E)$ be a bipartite graph. There exists a matching M of size $|S|$ if and only if:

$$\text{for every subset } A \subseteq S, \quad |N(A)| \geq |A|.$$

Here, $N(A)$ is the set of neighbours of A in B.

Hall's marriage theorem success

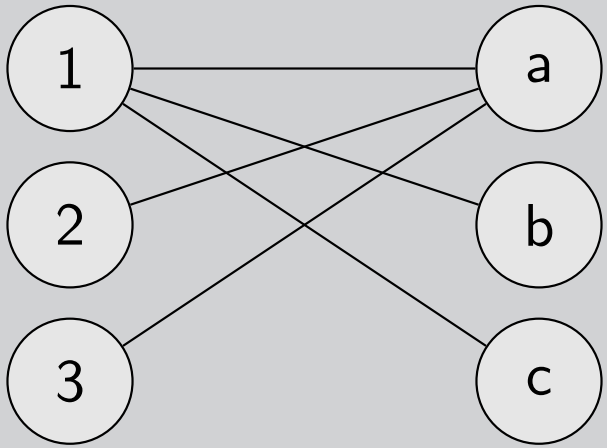
Example: Matching Exists



For any subset $A \subseteq \{1, 2, 3\}$, $|N(A)| \geq |A|$. E.g. $N(\{2, 3\}) = \{b, c, d\}$ and $|\{b, c, d\}| \geq |\{2, 3\}|$. Hence, a matching of size 3 exists.

Hall's Theorem Failure

Diagram



Conclusion

Here $N(\{1,2\}) = \{a\}$, and $|\{a\}| < |\{1,2\}|$, so Hall's condition fails. There is no matching of size 3.

Hall's Theorem: Maximal Matching Size

Quantitative Version

In a bipartite graph $G = (S \cup T, E)$, the size of a maximum matching is

$$|S| - \max_{A \subseteq S} (|A| - |N(A)|) .$$

Example

In the example on the previous slide, the maximum value of $|A| - |N(A)|$ is obtained by $A = \{2, 3\}$, for which $|A| - |N(A)| = 1$, which shows that the size of a maximum matching is $|S| - 1 = 2$.

König's theorem

A **vertex cover** of a graph $G = (V, E)$ is a subset $A \subseteq V$ such that every edge is incident some vertex in A.

Theorem

Let $G = (S \cup T, E)$ be a bipartite graph. Then the maximum matching of G equals the size of the smallest vertex cover of G.

Optimal matchings, weights on vertices

Setting

Let $G = (S \cup T, E)$ be a bipartite graph and $w\colon S \to \mathbb{R}$ a function. Want matching M in G that maximises $\sum_{(s,v)\in M} w(s)$.

Greedy Algorithm

1. Sort the vertices of S in decreasing order of weight, let $B = \varnothing$.
2. For each $s \in S$ (in this order):
   - If there is a matching in G whose S-vertices are $B \cup \{s\}$, then update B to $B \cup \{s\}$.
   - Otherwise, leave B as is.
   - Repeat step 2 with next element of S.

Optimal matchings, weights on vertices cont'd

Correctness

Let $\mathcal{B}$ be the set of subsets $B \subseteq S$ such that $B \in \mathcal{B}$ if B is the set of S-vertices of a matching. This makes $(S, \mathcal{B})$ into a matroid, so the greedy algorithm works.

Optimal matchings, weights on edges

Setting

Let G be a bipartite graph, and let $w\colon E \to \mathbb{R}$ be a function. The goal is to find a matching M with minimal weight

$$w(M) = \sum_{e \in M} w(e).$$

Special case

Assume further that $G = K_{n,n}$ and that w takes integer values. (General case can be reduced to this.) Then problem is: Given an integer $(n \times n)$-matrix C, find a **diagonal** D such that the sum of entries in D is minimal.

Hungarian algorithm

Given cost matrix C, modify it as follows to find a minimal diagonal D.

Main Steps (sketch)

1. Subtract minimum value of each row from each entry in the row.
2. Subtract minumum value of each column from each entry in the column.
3. Cover all zeros of C using the fewest number of lines (rows/columns).
   - If number of lines is $< n$: Let $\bar{w}$ be minimal entry not covered by line. Modify C by adding/subtracting $\bar{w}$ (details omitted).
   - If number of lines is n, there is a diagonal D containing only 0's. This is optimal.

Eulerian Circuits

Definition

An **Eulerian circuit** in a graph is a closed walk that uses every edge exactly once (but may repeat vertices). A graph is **Eulerian** if it contains such a circuit.

Theorem (Euler)

A connected multigraph has an Eulerian circuit if and only if every vertex has even degree.

Examples

- A cycle graph C_n is Eulerian.
- The "Königsberg bridges" graph is not Eulerian (vertices with odd degree).

Chinese postman problem

In a graph $G = (V, E)$, with weight function $w\colon E \to \mathbb{R}_{\geq 0}$, find a walk such that the sum of the weights of edges traveresed is minimal.

We're not doing this.

Hamiltonian Circuits

Definition

A **Hamiltonian circuit** in a graph is a cycle that visits every vertex exactly once and returns to the starting point. A graph that contains such a cycle is called **Hamiltonian**.

Examples

- The cycle graph C_n is Hamiltonian for all $n \geq 3$.
- The complete graph K_n is Hamiltonian for all $n \geq 3$.

Note

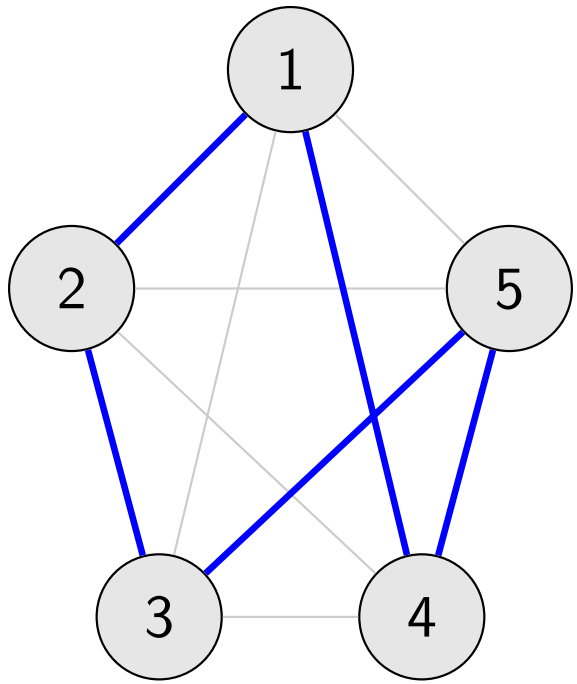
There is no simple necessary and sufficient condition for a graph to be Hamiltonian, and determining whether a graph has a Hamiltonian circuit is hard (specifically "NP-complete").

Example of a Hamiltonian Circuit

Hamiltonian Circuit in K_5

One Hamiltonian circuit is:

$$1 \to 2 \to 3 \to 5 \to 4 \to 1$$



Travelling Salesperson Problem (TSP)

Definition

Let $G = (V, E) = K_n$, and let $w \colon E \to \mathbb{R}_{\geq 0}$ be a weight function. The **Travelling Salesperson Problem (TSP)** asks for a Hamiltonian circuit $C \subseteq E$ such that:

$$\sum_{e \in C} w(e) \quad \text{is minimized.}$$

Metric TSP

Setting

Let $G = K_n$ with a weight function $w\colon E \to \mathbb{R}_{\geq 0}$ satisfying the **triangle inequality**:

$$w(x,z) \leq w(x,y) + w(y,z) \quad \text{for all } x,y,z \in V.$$

Solving the TSP under this assumption is called the **metric TSP**.

Mimimum spanning tree heuristic

Algorithm

1. Find a **minimum spanning tree** $T \subseteq K_n$ (Greedy alg.).
2. For every edge $e \in T$, add a copy of e to T to make a multigraph $\bar{T}$.
3. Find an Eulerian circuit C in $\bar{T}$.
4. Drop repeated vertices from C to get a Hamiltonian circuit in K_n.

Christofides' Heuristic for Metric TSP

For a metric weight function on K_n.

Heuristic

1. Compute a **minimum spanning tree** $T \subseteq K_n$.
2. Let $O \subseteq V$ be the set of vertices of odd degree in T.
3. Find a **minimum-weight maximum matching** M on the subgraph of K_n induced by O.
4. Form multigraph $\bar{T}$ by adding every edge in M to T.
5. Find an **Eulerian circuit** in $\bar{T}$ and drop repeated vertices to get a Hamiltonian circuit.

Approximation Guarantees

For a metric weighted K_n, let C_{opt}, C_{MST} and C_{CH} be the the Hamiltonian circuits which are optimal, produces by MST and produced by the Christofides heuristic, respectively. Then

$$w(C_{MST}) \leq 2w(C_{opt}) \text{ and } w_(C_{CH}) \leq \frac{3}{2}w(C_{opt}).$$

Algebraic systems

What Is a Field?

(Quasi-)definition

A **ring** is a set R equipped with operations of addition and multiplication, where "the usual arithmetic rules hold"; in particular there exists elements $0 \in R$ and $1 \in R$ such that

$$0 + x = x, 1x = x \quad \forall x \in R.$$

If for every non-zero $x \in R$, there is an element y (the **multiplicative inverse of** x, such that $xy = 1$, we say that R is a **field**.

Examples

- $\mathbb{Q}, \mathbb{R}, \mathbb{C}$: rational, real, complex numbers are fields.
- The integers $\mathbb{Z}$ is a ring, but not a field, since there is no $y \in \mathbb{Z}$ such that $2y = 1$.

Integers Modulo n

Definition

For a positive integer n, two integers a and b are **congruent modulo** n, written $a \equiv b \pmod{n}$, if n divides $a - b$.

Examples

- $17 \equiv 2 \pmod{5}$ since $17 - 2 = 15$ is divisible by 5.
- $-4 \not\equiv 4 \pmod{7}$ since $(-4) - 4 = -8$ is not divisible by 7.

Equivalence classes and $\mathbb{Z}_n$

Definition

Let $n \in \mathbb{N}$, $n \geq 1$. Define a relation $\sim_n$ on $\mathbb{Z}$ by:

$$a \sim_n b \quad \text{if and only if} \quad a \equiv b \pmod{n},$$

i.e., $n \mid (a - b)$. This is an equivalence relation.

Equivalence Classes

The set of all equivalence classes is $\mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z} = \{[0]_n, [1]_n, \ldots, [n-1]_n\}$, where

$$[a]_n = \{a + kn \mid k \in \mathbb{Z}\} = \{b \in \mathbb{Z} \mid b \sim_n a\}.$$

Equivalence classes and $\mathbb{Z}_n$, cont'd

Operations

The set $\mathbb{Z}_n$ has well-defined addition and multiplication operations given by, for $a, b \in \mathbb{Z}$:

$$[a]_n + [b]_n := [a+b]_n, \qquad [a]_n \cdot [b]_n := [ab]_n.$$

These satisfy all the usual rules of arithmetic, i.e. $\mathbb{Z}_n$ is a ring.

Notation

We usually omit $[]_n$ from the notation, e.g. writing 5 instead of $[5]_{12}$ for the congruence class containing 5 in $\mathbb{Z}_{12}$.

When Is $\mathbb{Z}_n$ a Field?

Theorem

The ring $\mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z}$ is a field (i.e. every element has a multiplicative inverse) if and only if n is a **prime number**.

Examples

- The number 4 is not prime, so $\mathbb{Z}_4$ is not a field. The element $2 \in \mathbb{Z}_4$ has no multiplicative inverse; there is no $y \in \mathbb{Z}$ such that $2y \equiv 1 \pmod 4$.
- Since 103 is prime and $5 \not\equiv 0 \pmod{1}03$, there is a unique integer $1 < y < 103$ such that $5y \equiv 1 \pmod{1}03$.

Fermat's Little Theorem

Theorem

Let p be a prime number and $a \in \mathbb{Z}$ be such that $p \nmid a$. Then

$$a^{p-1} \equiv 1 \pmod{p}.$$

Example

Let $p = 7$, and $a = 3$. Then:

$$3^6 = 729 = 7 \cdot 104 + 1 \equiv 1 \pmod{7}$$

Galois Fields

Theorem (Existence and Uniqueness)

For every prime number p and every integer $n \geq 1$, there exists a finite field with exactly p^n elements. This field is unique (up to isomorphism) and is denoted by $GF(p^n)$.

Galois fields of prime order

If $n = 1$, we have $GF(p) = \mathbb{Z}_p$.

Constructing Galois fields

Construction

- Let $\mathbb{Z}_p[x]$ be the set of polynomials with coefficients in $\mathbb{Z}_p$.
- To construct $\mathbb{F}_{p^n}$ find a polynomial $f(x) \in \mathbb{Z}_p[x]$ which is irreducible and of degree n.
- Say that $g(x), h(x) \in \mathbb{Z}_p[x]$ are congruent modulo $f(x)$ if $f(x)$ divides $g(x) - h(x)$.
- Then take

$$GF(p^n) = \mathbb{Z}_p[x]/(f(x)),$$

where $\mathbb{Z}_p[x]/(f(x))$ is the ring of congruence classes in $\mathbb{Z}_p[x]$ modulo $f(x)$.

Latin Squares

Definition

A **Latin square** of order n is an $n \times n$ array filled with n different symbols (usually $\{1, 2, \ldots, n\}$) such that:

- Each symbol appears exactly once in each row,
- Each symbol appears exactly once in each column.

More formally given by a function $L\colon \{1, \ldots, n\}^2 \to S$, where S is the set of symbols and $L(i, j)$ is the entry in spot (i, j) of the array.

Example: Latin Square of Order 3

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{bmatrix}$$

Orthogonal Latin Squares

Definition

Two Latin squares L_1 and L_2 of order n on symbol set S are said to be **orthogonal** if all the n^2 ordered pairs $(L_1(i,j), L_2(i,j)) \in S^2$ are distinct.

Example: Orthogonal Latin Squares of Order 3

$$L_1 = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{bmatrix} \quad L_2 = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \\ 2 & 3 & 1 \end{bmatrix} \quad (L_1, L_2) = \begin{bmatrix} (1,1) & (2,2) & (3,3) \\ (2,3) & (3,1) & (1,2) \\ (3,2) & (1,3) & (2,1) \end{bmatrix}$$

All entries in right hand square are distinct, so L_1 and L_2 are orthogonal.

Maximum Number of Orthogonal Latin Squares

Mutually Orthogonal Latin Squares (MOLS)

A set of Latin squares $L_1, \ldots, L_k$ of order n is called **mutually orthogonal** if every pair (L_i, L_j) with $i \neq j$ is orthogonal.

Maximal Number of MOLS

If $M(n)$ denotes the maximum number of MOLS of order n, then $M(n) \leq n - 1$ in general, and $M(n) = n - 1$ if n is a prime power.

Furter facts

- $M(n_1 n_2) \geq \min(M(n_1), M(n_2))$.
- $M(6) = 1$ (the Euler officer problem).

Construction of $p^k - 1$ MOLS

Construction

Let $q = p^k$, and let $GF(q) = \{a_1, a_2, \ldots, a_q\}$ with $a_q = 0$. Get $q - 1$ mutually orthogonal Latin squares $L_l \colon \{1, \ldots, q\}^2 \to GF(q)$ by setting

$$L_l(i, j) = a_l a_i + a_j \qquad 1 \leq l < q.$$

t-designs

Definition

A t-**design** with parameters v, k, λ consists of a pair $(S, \mathcal{B})$, where $\mathcal{B} \subseteq 2^S$, such that

- S is a set, $|S| = v$
- $\mathcal{B}$ is a set of k-subsets of S, called **blocks**
- for every $A \subseteq S$ with $|A| = t$, there are exactly λ blocks containing A.

Number of Blocks in a t-Design

Number of Blocks

The total number of blocks in a t- design with parameters v, k, λ is:

$$b = \lambda \cdot \frac{\binom{v}{t}}{\binom{k}{t}}.$$

Proof

Put together the following observations:

- There are $\binom{v}{t}$ t-subsets $A \subseteq S$.
- Each block contains $\binom{k}{t}$ such t-subsets.
- Each t-subset is contained in exactly λ blocks.

Projective Planes

Definition

A **projective plane** is a pair $(S, \mathcal{B})$ where S is a finite set, whose elements are called the **points**, and $\mathcal{B}$ is a set of subsets of S, called the **lines**. These must satisfy:

- Any two distinct points are contained in exactly one line.
- Any two distinct lines have exactly one point in common.
- There exist four points such that no three of them are contained in a line.

Order of a projective plane

Theorem

Every projective plane $(S, \mathcal{B})$ is a 2-design with $\lambda = 1$. There exists an n, called the **order** of the plane, such that

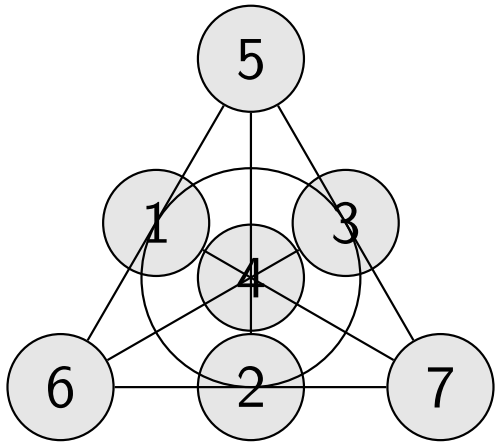
- The number of points $|S| = n^2 + n + 1$
- The number of lines $|\mathcal{B}| = n^2 + n + 1$
- Each line contains $k = n + 1$ points,
- Each point lies on $r = n + 1$ lines,

The Fano Plane

Definition

The **Fano plane** is the projective plane of order 2. It has:

- 7 points and 7 lines,
- Each line contains 3 points,
- Each point lies on 3 lines,
- Every pair of points lies on a unique line.



Existence of Projective Planes

Theorem

For every prime power $q = p^n$, there exists a finite projective plane of order q.

Construction

Such a plane can be constructed using linear algebra over finite fields:

- The points are 1-dimensional subspaces of $GF(q)^3$.
- The lines are 2-dimensional subspaces.
- Point contained in line if and only if corresponding 1d subspace contained in corresponding 2d subspace.

Existence of Projective Planes cont'd

Coordinate description

Points are equivalence classes of triples $[a_0 : a_1 : a_2] \in GF(q)^3 \setminus \{(0,0,0\}$ with equivalence relation

$$[a_0 : a_1 : a_2] \sim [b_0 : b_1 : b_2]$$

if there is a $\lambda \in GF(q) \setminus \{0\}$ such that $\lambda a_i = b_i$ for all i. The unique line containing distinct points p_1 and p_2 is the set

$$\langle p_1, p_2 \rangle = \{\lambda_1 p_1 + \lambda_2 p_2 \mid (\lambda_1, \lambda_2) \in GF(q)^2 \setminus \{(0,0)\}\}$$

Open Question

It is unknown whether projective planes exist for any orders that are *not* prime powers.

Introduction to Coding Theory

Goal

Coding theory studies how to transmit information over noisy channels in such a way that errors can be detected and corrected.

Formalities

Given is a finite set X (e.g. $X = \{A, B, \ldots, Z\}$) as source alphabet, and a finite set A (e.g. $A = \{0, 1\}$) as encoding alphabet. Let $A^* = \bigcup_{n \geq 0} A^n$ be the set of all words in alphabet A.

- A **source encoding** is a function $c \colon X \to A^*$.
- For $x \in X$, the output $c(x) \in A^*$ is called a **code word**
- The set $c(X) \subseteq A^*$ is called the **code**.

Shannon's Noisy Channel Coding Theorem

Setup

Given a communication channel with some statistical rule describing how symbols are corrupted in transmission. There exists a number C with $0 \leq C \leq 1$ called the **channel capacity**.

Shannon's Theorem (Noisy Channel)

For any rate $R < C$ and $\epsilon > 0$, there exists a coding scheme that transmits data at rate R with error rate less than ϵ.

Interpretation

Shannon's theorem guarantees the existence of "good" codes, but does give explicit ways to construct them.

Prefix Codes

Definition

A **prefix code** is a set of strings (codewords) over a finite alphabet A such that no codeword is a prefix of another codeword.

Examples

With $A = \{0, 1\}$:

- The code $\{0, 10, 110\}$ is prefix.
- The code $\{0, 01\}$ is not prefix, since 0 is a prefix of 01

Application

- Ensures an X-word $x_1 \dots x_n$ encoded to an A-word $c(x_1) \dots c(x_n)$ can be decoded uniquely.
- Used in data compression (e.g., Huffman coding).

Block Codes

Definition

A **block code** of length n over an alphabet A is a code C where every codeword has length n, i.e.

$$C \subseteq A^n,$$

Block code examples

Example: Parity Code (even parity)

Code over $A = \{0, 1\}$:

$$C = \{x \in \{0,1\}^n \mid \text{the number of 1s in } x \text{ is even}\}$$

Example: Double Repetition Code

Codewords of the form $(x, x, x) \in A^{3n}$, where $x \in A^n$:

$$C = \{(x_1, \ldots, x_n, x_1, \ldots, x_n, x_1, \ldots, x_n) \mid x_i \in A\}$$

Hamming Distance

Definition

Let $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ be two words in A^n. The **Hamming distance** between x and y is:

$$\Delta(x, y) = |\{i \in \{1, \ldots, n\} \mid x_i \neq y_i\}|.$$

Interpretation

This is a metric, that is:

- $\Delta(x, y) \geq 0$
- $\Delta(x, y) = 0 \iff x = y$,
- $\Delta(x, y) = \Delta(y, x)$
- $\Delta(x, z) \leq \Delta(x, y) + \Delta(y, z)$ (triangle inequality).

Error Detection and Correction

Distance of a Code

Let $C \subseteq A^n$ be a block code. The **distance** d of C is:

$$d = \min\{\Delta(x, y) \mid x, y \in C,\ x \neq y\}.$$

Error detection and correction

Let $t \geq 0$, then we have that C is

- t-**error detecting** if $d(C) \geq t + 1$.
- t-**error correcting** if $d(C) \geq 2t + 1$.

Hamming Bound

Hamming Bound (Sphere-Packing Bound)

Let $C \subseteq A^n$, let $|A| = q$, and assume that C is t-error correcting. Then

$$|C| \leq \frac{q^n}{\sum_{i=0}^{t} \binom{n}{i}(q-1)^i}.$$

Interpretation

Denominator is the number of words within Hamming distance $\leq t$ of a given codeword. These balls must fit inside the space A^n of size q^n without overlapping.

t-Perfect Codes

Definition

A code $C \subseteq A^n$ is called t-**perfect** if it is t-error correcting achieves equality in the Hamming bound:

$$|C| = \frac{q^n}{\sum_{i=0}^{t} \binom{n}{i}(q-1)^i}.$$

A t-perfect code partitions the entire space A^n into disjoint Hamming balls of radius t centered at the codewords.

Linear Codes

Definition

Let $A = F = GF(q)$, a finite field. A **linear code** of length n over F is a linear subspace:

$$C \subseteq F^n.$$

Linear subspace axioms

This means

- $\vec{0} \in C$ contains the zero vector
- $x, y \in C \Rightarrow x + y \in C$
- $x \in C, \lambda \in F \Rightarrow \lambda x \in C$.

Properties of linear codes

- A linear code $C \subseteq F^n$ has a basis $\{x_1, \ldots, x_k\} \subseteq C$, i.e. a set of linearly independent vectors spanning C.
- The cardinality k of the basis is the dimension of C.
- We have $|C| = q^{\dim C}$.

Minimum Distance and Minimum Weight

Weight of a Codeword

The **weight** $w(x)$ of a codeword $x \in F^n$ is the number of nonzero coordinates in x (i.e. $w(x) = \Delta(x, 0)$).

Fact

If C is a linear code, then:

$$d(C) = \min\{w(x) \mid x \in \mathcal{C},\ x \neq 0\}.$$

The distance equals the **smallest weight of a nonzero codeword**.

Generator Matrices

Definition

A **generator matrix** for a linear code $C \subseteq F^n$ of dimension k is a $(k \times n)$-matrix G with entries in F whose rows form a basis for C.

Encoding

Any codeword $c \in C$ can be written as:

$$c = xG, \quad \text{for some } x \in F^k.$$

Example

The parity code of length 3 has generator matrix

$$G = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

Dual Codes

Definition

Let $C \subseteq F^n$ be a linear code. The **dual code** of C, denoted $C^\perp$, is defined by:

$$C^\perp = \{y \in F^n \mid x \cdot y = 0 \text{ for all } x \in C\}.$$

Basic Facts

- If $x_1, \ldots, x_k$ is a basis of C, then

$$y \in C^\perp \iff x_i \cdot y = 0 \quad \forall i.$$

- $C^\perp$ is a linear code.
- $\dim(C^\perp) = n - \dim(C)$.
- $(C^\perp)^\perp = C$.

Parity-Check Matrices

Definition

A **parity-check matrix** H for a linear code C is a generator matrix for the dual code $C^{\perp}$.

Usage

A vector $x \in F^n$ lies in C if and only if:

$$Hx = 0.$$

In other words, C is the null space of H.

Minimum Distance via Parity-Check Matrix

Key Fact

The distance d of $\mathcal{C}$ is the smallest number of columns of H that are linearly dependent:

$$d = \min\{t \mid \text{some } t \text{ columns of } H \text{ are linearly dependent}\}.$$

Example

The parity code of length 3 has

$$H = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix},$$

where any two column vectors are linearly dependent, and every single column vector is nonzero. Thus $d(C) = 2$.

Hamming Codes

Definition/construction

- Hamming codes are 1-perfect codes.
- Defined by specifying a parity check matrix H with l rows and $(l^q - 1)/(q - 1)$ columns, such that every non-zero vector in F^l appears once (up to rescaling) as a column vector.

Example

Taking $q = 2, l = 3$ gives

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Syndromes in Linear Codes

Definition

Let $C \subseteq F^n$ be a linear code with parity-check matrix $H \in F^{(n-k)\times n}$. For any received word $r \in F^n$, the **syndrome** is:

$$s = Hr^\top \in F^{n-k}.$$

Properties

- $s = 0$ if and only if $r \in C$,
- If $r = c + e$ for $c \in C$, then:

$$s = Hr^\top = H(c + e) = He.$$

- If C is t-error correcting and $w(e) \leq t$, can determine e from s.

Cyclic Codes

Definition

A linear code $C \subseteq F^n$ is called a **cyclic code** if:

$$(a_1, a_2, \ldots, a_n) \in C \quad \Rightarrow \quad (a_2, a_3, \ldots, a_{n-1}, a_1) \in C.$$

That is, C is closed under cyclic shifts of codewords.

Polynomial Representation

Every word $c = (c_{n-1}, \ldots, c_0) \in F^n$ corresponds to a polynomial:

$$c(x) = c_{n-1}x^{n-1} + c_{n-2}x^{n-2} + \cdots + c_0.$$

Generator Polynomial

Theorem

A cyclic code $C \subseteq F^n$ is uniquely determined by a monic polynomial $g(x)$ which divides $x^n - 1$, such that we have

$$g(x) \mid f(x) = a_{n-1}x^{n-1} + \cdots + a_0 \iff (a_{n-1}, \ldots, a_0) \in C.$$

Construction

The generator matrix G has rows:

$$\text{row}_i = x^i g(x), \quad \text{for } i = 0, \ldots, n - \deg g - 1,$$

interpreted as coefficient vectors in F^n.

Cyclic code example

Example

Let $g(x) = x^3 + x + 1 \in \mathbb{F}_2[x]$, $n = 7$. Then $\deg g = 3$, so $k = 4$. The generator matrix is

$$G = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Verification via Polynomial Division

Key Fact

Let $C \subseteq F^n$ be a cyclic code with generator polynomial $g(x) \mid x^n - 1$.
A word $c(x) \in F[x]$ of degree $< n$ is a codeword in C if and only if:

$$g(x) \mid c(x).$$

Verification Procedure

To check whether a received word $r(x) \in F^n$ is a valid codeword:

1. Compute the remainder of $r(x)$ divided by $g(x)$,
2. If the remainder is 0, then $r(x) \in C$,
3. Otherwise, an error has occurred.

Basics of Cryptography

Goal

Alice and Bob want to communicate between themselves in such a way that the other can understand them, but eavesdropper Eve can not understand what is said.

Mathematical formulation

Specify three sets T, C, K and two functions $c\colon T \times K \to C$, $d\colon C \times K \to T$, where

- T is the set of possible messages
- C is the set of possible encrypted messages
- K is a set of "keys"
- c and d represent encryption and decryption, respectively.

Primitive Roots Modulo p

Definition

Let p be a prime number. A residue class $g \in \mathbb{Z}_p$ is called a **primitive root modulo** p if every non-zero element of $\mathbb{Z}_p$ can be written as a power of g.

Key Fact

For every prime p there exists a primitive root.

Example

For $p = 7$, the element $g = 3$ is a primitive root:

$$3^1 \equiv 3,\ 3^2 \equiv 2,\ 3^3 \equiv 6,\ 3^4 \equiv 4,\ 3^5 \equiv 5,\ 3^6 \equiv 1 \pmod 7$$

The Discrete Logarithm Problem (DLP)

Problem Statement

Let p be a prime, and let $g \in \mathbb{Z}_p \setminus \{0\}$ be a primitive root. Given g and $h \in \mathbb{Z}_p^*$, find an integer $x \in \{0, \ldots, p-2\}$ such that:

$$g^x \equiv h \pmod{p}.$$

This x is called the **discrete logarithm of h to the base g** modulo p.

For large primes p, computing discrete logarithms is computationally hard.

Example

In $\mathbb{Z}_7$, with $g = 3$, $3^4 \equiv 4 \pmod 7$, so $\log_3 4 = 4$. This is computable since $p = 7$ is small.

Diffie–Hellman Key Exchange

Public Setup

- A large prime p, and a primitive root g modulo p, are publicly agreed upon.

Key Exchange Protocol

- Alice chooses a secret $a \in \{1, \ldots, p-2\}$ and sends $A = g^a \mod p$ to Bob.
- Bob chooses a secret $b \in \{1, \ldots, p-2\}$ and sends $B = g^b \mod p$ to Alice.
- Alice computes $K = B^a \mod p$.
- Bob computes $K = A^b \mod p$.

Since $B^a \equiv g^{ba} \equiv g^{ab} \equiv A^b \mod p$, both parties now share the same secret key K.

Diffie–Hellman security

Security

An eavesdropper sees g, p, A, B, but cannot compute $K = g^{ab} \mod p$ without solving the discrete logarithm problem.

RSA Encryption Scheme

Alice wants to receive messages safely from anyone.

Key Generation (by Alice)

- Choose two large secret primes p and q.
- Compute $n = pq$ and $(p-1)(q-1)$.
- Choose $e \in \mathbb{Z}$ coprime to n (typically small, e.g. $e = 65537$).
- Compute $d \in \mathbb{Z}$ such that $ed \equiv 1 \mod \varphi(n)$.

The public key is (n, e), the private key is d.

RSA II

Encryption and Decryption

- For Bob to encrypt a message $m \in \mathbb{Z}_n$, computes:

$$c = m^e \mod n.$$

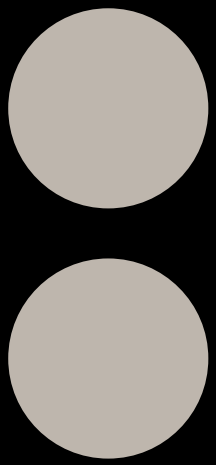
- For Alice to decrypt the ciphertext c, computes:

$$m = c^d \mod n.$$

Security

Based on the assumption that factoring $n = pq$ is hard. Even knowing n and e, computing d requires factoring n, which is computationally infeasible if p and q are large enough.

UiO : Department of Mathematics
University of Oslo



MAT3250 Spring 2025

Discrete mathematics revision

