Chapter 1

Graph theory

simple	undirected	finite	deterministic	unweighted
multigraph	directed	infinite	probabilistic	weighted

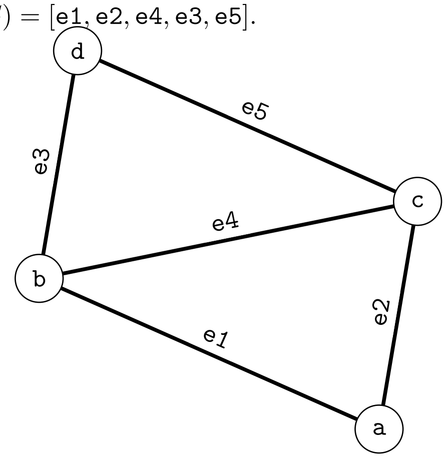
1.1 Basic definition

Definition 1.1.1 (graph)**.** *The following are various definitions of graphs*

1. *A* graph *G is a non-empty set, $V(G)$, of objects, called* vertices*, together with a set, $E(G)$, of unordered pairs of (distinct) vertices. The elements of $E(G)$ are called* edges*;*

2. *A* graph *G is a triple consisting of non-empty set $V(G)$, of objects, called* vertices*, a set, $E(G)$, of objects called* edge *and a relation that associates each edge with a pair of vertices;*

Example $V(G) = [\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}]\,, E(G) = [\mathtt{e1}, \mathtt{e2}, \mathtt{e4}, \mathtt{e3}, \mathtt{e5}]$.



We use the following terminology:

- if edge $e = (u, v)$ then we say vertices u and v are *adjacent*;
- if edge $e = (u, v)$ then we say edge e is *incident* with u and v;

- if edge $e = (u, v)$ then we say edge e is *joins* u and v;
- vertices adjacent to a vertex u are *neighbours* of u; the set of neighbours of u is denoted by $N(u)$;
- if an edge joins vertex with itself we call it a *loop*;
- if there are more than one edge incident with the same pair of edges then the graph has *multiple edges*;
- we write $uv \in G$ if G has edge incident with u and v;

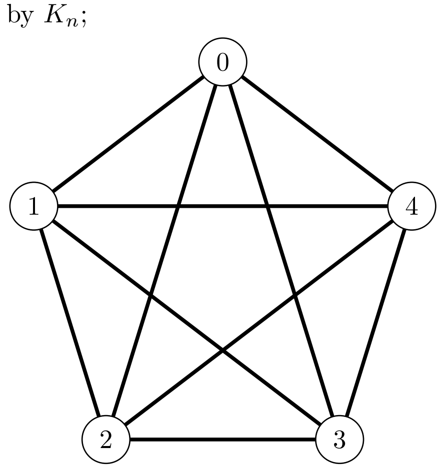
Definition 1.1.2 (simple graphs)**.** *A graph is called* simple *if it has no loops and no multiple edges.*

Definition 1.1.3 (finite graphs)**.** *A graph is called* finite *if both the edge set and the vertex sets are finite; otherwise the graph is called* infinite.

A simple graph can be both finite and infinite. Those two ideas are independent from each other.

- we call a graph G empty if the set of edges is empty; an empty graph with n edges is typically denoted by E_n; for example E_4 is

- we call a simple graph G *complete* if every pair of vertices in the graph are adjacent; an complete graph with n edges is typically denoted by K_n;



We have the following terminology

- a *walk* from vertex a to vertex b in a graph G is an alternating sequence $v_0e_1v_1e_2v_2\dots e_nv_n$ of vertices and edges in G such that $a = v_0$, $b = v_n$ and for all $1 \le i \le n$ edge $e_i = (v_{i-1}v_i)$;
- the *length of a walk* $v_0e_1\dots v_n$ is the number of edges that appear in the walk sequence;
- a *trail* is a walk with no repeated edges;
- a *path* is a walk with no repeated vertices;
- a *closed* walk is walk for which $v_0 = v_n$;
- a *circuit* is a closed trail;
- a *cycle* is a closed walk such that vertices $v_0, \dots, v_{n-1}$ are all distinct; note that the index of vertices reaches $n-1$ not n. As noted above for closed walks $v_0 = v_n$.

Definition 1.1.4 (girth)**.** *The girth of a graph is the length of a cycle of the shortest length. If the graph has no cycles we will say its girth is positive infinity.*[1]

For graph G depicted in the first example we have girth 3. For the complete graph on five vertices the girth is 3. See bellow for more examples.

Theorem 1.1.1. *If there is a walk from a to b in G then there is a path from a to b in G.*

Theorem 1.1.2. *If there is a path from a to b and from b to c in G then there is a path from a to c in G.*

Definition 1.1.5 (degree)**.** *The number of edges incident with a vertex u is called the* degree *of u and denoted by* $\deg(u)$.

Definition 1.1.6 (degree sequence)**.** *The degree sequence of a graph is the list of vertex degrees usually listed in a decreasing order as $d_1 \ge d_2 \ge \dots \ge d_n$*

The minimum degree of a graph is denoted by $\delta(G)$ and the maximum degree of a graph via $\Delta(G)$. A graph where every vertex has degree k is called k-regular graph.

Theorem 1.1.3 (Handshake lemma)**.** *For a graph G having e edges*

$$\sum_{u \in V(G)} d(v) = 2e$$

Proof. Consider the collection of unordered pairs (u, v) where each pair corresponds to an edge in the graph. There are in total $2e$ vertices counted with multiplicities in the list. Another way to count the same value is look at how many times a vertex appears in a pair (u, v), than answer is the degree of the pair. Hence the result follows. □

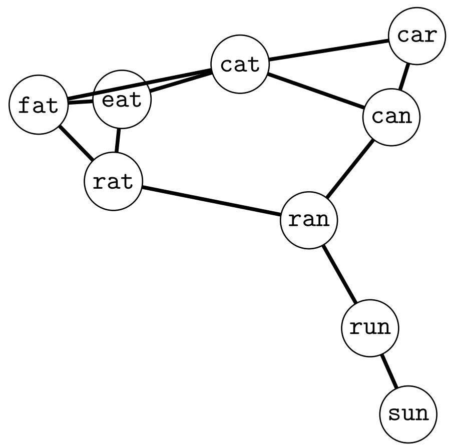
Question: Can we have a graph with 3 vertices of degree one and all other vertices of degree two?

Theorem 1.1.4. *The number of vertices of odd degree in a graph is even*

[1] There may be alternative definitions when there are no cycle, in this class however we will use $+\infty$

1.2 Graph representation

Example Let WG be the set of the words in [`can`, `car`, `cat`, `eat`, `fat`, `ran`, `rat`, `run`, `sun`]. Two words are adjacent if and only if they differ in exactly on position by exactly one letter. The drawing of the graph is given bellow.



One important issue is how do we represent a graph in a computer.

Adjacency list: One way to store a graph is to keep for each vertex u a list of vertices that are adjacent to u (the *neighbourhood* of u). For the graph above we have the list

1. `can` $\rightarrow$ [`ran`, `car`, `cat`]
2. `car` $\rightarrow$ [`cat`, `can`]
3. `cat` $\rightarrow$ [`car`, `fat`, `can`, `eat`]
4. `eat` $\rightarrow$ [`fat`, `cat`, `rat`]
5. `fat` $\rightarrow$ [`cat`, `rat`, `eat`]
6. `ran` $\rightarrow$ [`run`, `rat`, `can`]
7. `rat` $\rightarrow$ [`ran`, `fat`, `eat`]
8. `run` $\rightarrow$ [`ran`, `sun`]
9. `sun` $\rightarrow$ [`run`]

Definition 1.2.1 (adjacency matrix)**.** *Let G be a graph with vertex set $V(G) = \{v_1, \ldots, v_n\}$. The* adjacency matrix *of G is a $n \times n$ matrix $A = \{a_{ij}\}$ where $a_{ij} = 1$ if v_i and v_j are adjacent and zero otherwise.*

An adjacency matrix of the graph above is

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

The adjacency matrix of a simple graph is symmetric. Furthermore depending on the labeling of the rows and columns a graph may have more than one adjacency matrix.

Definition 1.2.2 (Complement of a graph)**.** *Let G be a graph with edge set E and vertex set V. The* complement *of G is the graph $\overline{G} = (V, E')$ where $e = \{u, v\} \in E'$ if and only if vertices u and v are not adjacent in G.*

Observe that if the graph is dense it is better to store the complement of the graph rather than the graph itself. In other words in the adjacency list keep the vertices that are *not* adjacent to the vertex u.

Comment: Consider the adjacency matrix of a graph, the adjacency matrix of the complement flips zeroes and ones off the main diagonal. Thus to store a *dense* graph (a graph is dense it if has too many ones in the adjacency matrix) it is more efficient to store the complement of the graph.

Definition 1.2.3 (incidence matrix)**.** *Let G be a graph with vertex set $V(G) = \{v_1, \ldots, v_n\}$ and edge set $E(G) = \{e_1, \ldots, e_k\}$. The* incidence matrix *of G is a $n \times k$ matrix $B = \{b_{ij}\}$ where $b_{ij} = 1$ if v_i is incident with e_j and zero otherwise.*

For the graph in the beginning of this section an incidence matrix is given by

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Theorem 1.2.1. *For a graph G with adjacency matrix A and incidence matrix B we have*

$$BB^T = A + \begin{pmatrix} d(v_1) & 0 & \ldots & 0 \\ 0 & d(v_2) & \ldots & 0 \\ \vdots & & & \\ 0 & 0 & \ldots & d(v_n) \end{pmatrix}$$

Proof. Homework ∎