Discrete mathematics 1.

Graphs

Juhász Zsófia
jzsofia@inf.elte.hu
jzsofi@gmail.com
Based on Hungarian slides by Mérai László

Department of Computer Algebra

Spring 2019

Basic concepts of graphs

Definition (graph (undirected))

A triple $G = (\varphi, E, V)$ is called an (undirected) graph, if E and V are sets such that $V \neq \emptyset$, $V \cap E = \emptyset$ and $\varphi\colon E \to \{\{v, v'\} \,|\, v, v' \in V\}$. E is called the set of edges, V is called the set of vertices (nodes) and φ is the incidence map. (The map φ assigns to each element of E an unordered pair of elements in V.)

If $v \in \varphi(e)$ then we say that e is incident to v and v is an endpoint of e.

Comment

The incidence map determines the so called incidence relation $I \subseteq E \times V$: $(e, v) \in I \Leftrightarrow v \in \varphi(e)$.

Basic concepts of graphs

Definition (finite graphs and empty graphs)

If E and V are both finite sets, then the graph is called a finite graph, otherwise it is an infinite graph.
If $E = \emptyset$ then the graph is called an empty graph.

Most graphs considered in informatics are finite, therefore in the rest of this course we are going to study finite graphs.

Definition (loops, parallel edges, simple graphs)

If an edge is incident only to one vertex then we call this edge a loop.
If $e \neq e'$ and $\varphi(e) = \varphi(e')$ then e and e' are called parallel edges.
If a graph does not contain any loops nor any parallel edges, then this graph is called a simple graph.

Basic concepts of graphs

Definition (incident edges, adjacent vertices)

The edges $e \neq e'$ are called incident (or in some sources adjacent), if there exists $v \in V$ such that $v \in \varphi(e)$ and $v \in \varphi(e')$. The vertices $v \neq v'$ are adjacent, if there exists $e \in E$ such that $v \in \varphi(e)$ and $v' \in \varphi(e)$.

Definition (degree of a vertex)

The degree of a vertex v is the number of edges incident to it, counting each loop twice. Notation: $d(v)$ or $deg(v)$.

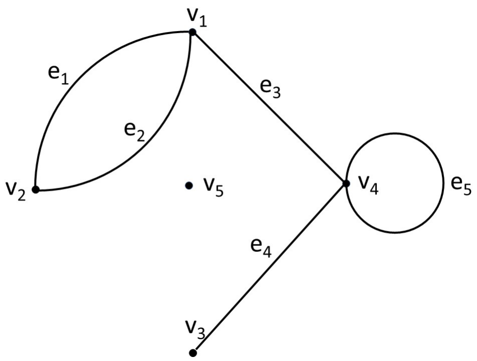
Definition (isolated vertex)

If $d(v) = 0$ for some $v \in V$ then v is called an isolated vertex.

Definition (regular graphs)

If the degree of every vertex in a graph is equal to n for some $n \in \mathbb{N}$ then the graph is called n-regular. A graph is called regular, if it is n-regular for some $n \in \mathbb{N}$.

Example



$V = \{v_1, v_2, v_3, v_4, v_5\}$
$E = \{e_1, e_2, e_3, e_4, e_5\}$
$\varphi = \{(e_1, \{v_1, v_2\}), (e_2, \{v_1, v_2\}), (e_3, \{v_1, v_4\}), (e_4, \{v_3, v_4\}), (e_5, \{v_4\})\}$

The sum of the degrees of all vertices in a graph

Theorem (The sum of the degrees of all vertices in a graph)

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

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

Proof

Induction by the number of edges in the graph:

Base step: If $|E| = 0$ then the values on both sides of the equality are 0.

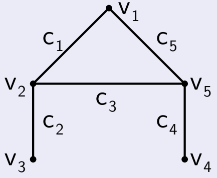
Inductive step: Assume that the statement is true when $|E| = n$, for some $n \in \mathbb{N}$. Let G be a graph with $n + 1$ edges. By deleting one edge of G we obtain a graph G' with n edges. By our inductive hypothesis, the statement is true for G'. If we now add to G' the edge deleted earlier from G, the values on both sides of the equality will increase by 2, hence the equality remains true. Therefore the statement also holds for G.

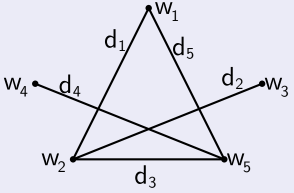
Basic concepts of graphs

Definition (isomorphic graphs)

Two graphs $G = (\varphi, E, V)$ and $G' = (\varphi', E', V')$ are said to be isomorphic to each other, if there exist bijections $f \colon E \to E'$ and $g \colon V \to V'$ such that for every $v \in V$ and $e \in E$, v is an endpoint of e if and only if $g(v)$ is an endpoint of $f(e)$.

Example





Suitable bijections f and g:
$f = \{(c_1, d_1), (c_2, d_2), (c_3, d_3), (c_4, d_4), (c_5, d_5)\}$
$g = \{(v_1, w_1), (v_2, w_2), (v_3, w_3), (v_4, w_4), (v_5, w_5)\}$

Basic concepts of graphs

Definition (deleting edges)

Let $G = (\varphi, E, V)$. The graph obtained from G by deleting the set of edges $E' \subseteq E$ is the graph $G' = (\varphi|_{E \setminus E'}, E \setminus E', V)$.

Definition (deleting vertices)

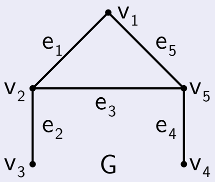
Let $G = (\varphi, E, V)$ be a graph and $V' \subseteq V$. Denote by E' the set of those edges which are incident to some vertex in V'. The graph obtained from G by deleting the set of vertices V' is the graph $G' = (\varphi|_{E \setminus E'}, E \setminus E', V \setminus V')$.

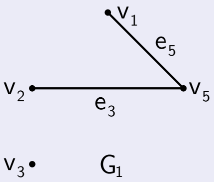
Basic concepts of graphs

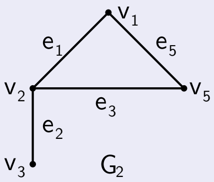
Definition (subgraphs, supergraphs)

A graph $G' = (\varphi', E', V')$ is called a subgraph of a graph $G = (\varphi, E, V)$, if $E' \subseteq E$, $V' \subseteq V$ and $\varphi' \subseteq \varphi$. In this case we also say that G is a supergraph of G'.
If E' contains all those edges in V which have both endpoints in V', then G' is called a subgraph spanned (or induced) by V'.

Example







G_1 is a subgraph, but not a spanned subgraph of G and G_2 is a spanned subgraph of G.

Basic concepts of graphs

Definition (complete graphs)

A simple graph in which every vertex is adjacent to all other vertices is called a complete graph. The complete graph with $n \in \mathbb{N}^+$ vertices is denoted by K_n.

Comments

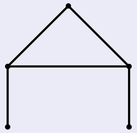
- It is easy to show that for any $n \in \mathbb{N}^+$, all complete graphs on n vertices are isomorphic, hence K_n is unique up to graph isomorphism.
- For every $n \in \mathbb{N}^+$, K_n has $\binom{n}{2} = n(n-1)/2$ edges.

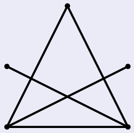
Basic concepts of graphs

Definition (complement of a simple graph)

The complement $\overline{G}$ of a simple graph G is a simple graph on the same set of vertices as G in which two vertices are adjacent if and only if they are not adjacent in G.

Example


G


$\overline{G}$

Basic concepts of graphs

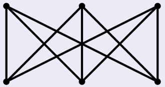
Definition (bipartite graphs)

A graph $G = (\varphi, E, V)$ bipartite graph, if V can be expressed as the union of two disjoint sets V' and V'' such that for every edge e in E one endpoint of e is in V' and the other endpoint of e is in V''.

Definition (the graphs $K_{m,n}$)

Let $m, n \in \mathbb{N}^+$. The simple bipartite graph in which $|V'| = m$ and $|V''| = n$ and every vertex in V' is adjacent to every vertex in V'', is denoted by $K_{m,n}$.

Example

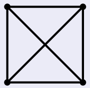

$K_{3,3}$

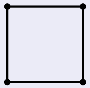
Further special graphs

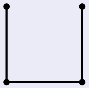
Definition (cycle graphs, path graphs, stars)

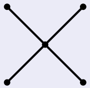
- For any $n \in \mathbb{N}^+$ a cycle graph C_n (or n-cycle or n-gon) is a graph with n vertices corresponding to the vertices of an n-sided polygon and two vertices in C_n are adjacent if and only if they are adjacent in the polygon.
- For any $n \in \mathbb{N}$ a path graph P_n is the graph obtained by deleting one edge from C_{n+1}.
- For any $n \in \mathbb{N}^+$, the graph $K_{n,1}$ is also called a star graph and is denoted by S_n.

Examples


K_4


C_4


P_3


S_4

Basic concepts of graphs

Definition (walk)

Let $G = (\varphi, E, V)$ be a graph, $n \in \mathbb{N}$. A walk of length n from vertex v_0 to v_n is a sequence:

$$v_0, e_1, v_1, e_2, v_2, \dots, v_{n-1}, e_n, v_n$$

where

- $v_j \in V \quad 0 \leq j \leq n$,
- $e_k \in E \quad 1 \leq k \leq n$,
- $\varphi(e_m) = \{v_{m-1}, v_m\} \quad 1 \leq m \leq n$.

If $v_0 = v_n$, then the walk is a closed walk, otherwise it is an open walk.

Definition (trail)

If a walk does not contain repeated edges, then it is called a trail (or line). According to the above definition, a trail can be an open trail or a closed trail.

Basic concepts of graphs

Definition (path)

If a walk does not contain repeated vertices then it is called a path.

Comments

- Every path is also a trail.
- A walk of length 0 is also a path consisting of a single vertex.
- A walk of length 1 is a path if and only if the single edge contained by it is not a loop.

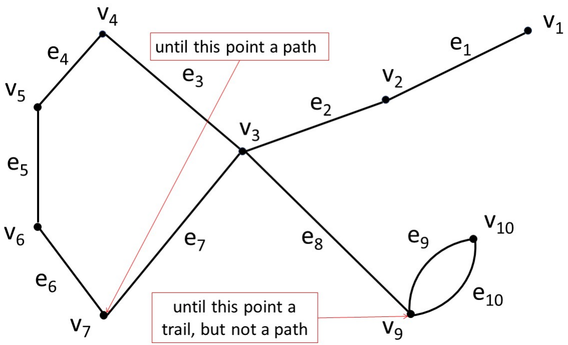
Definition (cycle)

A cycle is a closed trail of length ≥ 1 which contains no repeated vertices apart from the first and last vertices, which are identical.

Definition (circuit)

A circuit is a closed trail of length ≥ 1.

Example



path: $v_1, e_1, v_2, e_2, v_3, \dots, v_6, e_6, v_7$;
trail but not a path: $v_1, e_1, v_2, e_2, v_3, \dots, v_7, e_7, v_3, e_8, v_9$;
cycle: $v_3, e_3, v_4, e_4, v_5, e_5, v_6, e_6, v_7, e_7, v_3$.

Summary

walk	vertices may repeat	edges may repeat	closed or open
trail	vertices may repeat	edges cannot repeat	closed or open
path	vertices cannot repeat	edges cannot repeat	open (except for the trivial path)
circuit	vertices may repeat	edges cannot repeat	closed
cycle	first and last vertices are the same, other vertices cannot repeat	edges cannot repeat	closed

Basic concepts of graphs

Proposition (Creating a path from a walk)

Given any walk between two distinct vertices v and v' of a graph, we can obtain a path from v to v' by deleting suitable vertices and edges of the walk.

Proof

Consider the walk:

$$v = v_0, e_1, v_1, e_2, v_2, \dots, v_{n-1}, e_n, v_n = v'.$$

If this walk does not contain any repeated vertices, then it is already a path. Otherwise we have $v_i = v_j$ for some $i < j$. By deleting the part

$$e_{i+1}, v_{i+1}, e_{i+2}, v_{i+2}, \dots, v_{j-1}, e_j, v_j$$

from the walk we obtain a shorter walk from v to v'. Repeat this step until there are no repeated vertices in the walk. The process will finish in finite number of steps, since the length of the walk reduces in each step. When the process stops there are no repeated vertices, hence we obtained a path.

Basic concepts of graphs

Definition (connected graphs)

A graph is said to be connected if there is a walk between any pair of vertices of the graph.

For a graph $G = (\varphi, E, V)$ define the relation $\sim$ on V: let $v \sim v'$ if and only if there exists a walk from v to v' in G.

Since $\sim$ is an equivalence relation (Why?), the set of corresponding equivalence classes will be a partition of V.

The subgraphs spanned by these equivalence classes are called the components of the graph.

Comment

A graph is connected if and only if it consists of only one component.

Trees

Definition (tree)

A graph is called a tree if it is connected and contains no cycle (in other words: it is an *acyclic* graph).

Theorem (Equivalent characterisations of trees)

For a simple graph G the following conditions are equivalent:

1. *G is a tree;*
2. *G is connected, and by deleting any of its edges, the remaining subgraph is not connected (i.e. G is a minimally connected graph);*
3. *there is exactly one path between any vertices v and v' in G;*
4. *G contains no cycles, but by adding any new edge to G, the new graph will contain a cycle (i.e. G is a maximally acyclic graph).*

Structure of the proof

$(1) \Rightarrow (2) \Rightarrow (3) \Rightarrow (4) \Rightarrow (1)$

Trees

Proof

(1) $\Rightarrow$ (2) : *By the definition of a tree, G is connected. The other part of the statement we show by proof by contradiction.*
Suppose there is an edge e (denote its endpoints by v and v') in the graph such that after deleting e the remaining subgraph is connected. Then in the remaining subgraph there is a path from v to v'. Adding e and v to this path we obtain a cycle in G:
$v, e_1, v_1, e_2, \ldots, v_{n-1}, e_n, v', e, v.$ ↯

(2) $\Rightarrow$ (3) : *Let v and v' be vertices in G. Since G is connected, there is at least one path from v to v'. We show that there cannot exist two different paths from v to v', by proof by contradiction:*
Suppose there exist 2 different paths from v to v':
$v, e_1, v_1, e_2, \ldots, v_{n-1}, e_n, v'$ *and* $v, e'_1, v'_1, e'_2, \ldots, v'_{m-1}, e'_m, v'$. *Let k be the smallest index such that $v_k \neq v'_k$. (Why does such a k exist?) By deleting the edge e_k from G we obtain a connected subgraph, because the walk v_{k-1}, e_k, v_k can be replaced by the walk*
$v_{k-1}, e'_k, v'_k, \ldots, e'_m, v', e_n, v_{n-1}, e_{n-1}, v_{n-2}, \ldots, v_{k+1}, e_{k+1}, v_k.$ ↯

Trees

Proof

$(3) \Rightarrow (4)$: *We show that* G *contains no cycle, by proof by contradiction: Suppose there is a cycle* $v, e_1, v_1, e_2, \ldots, v_{n-1}, e_n, v$ *in* G*. Then there exist two different paths between* v_1 *and* v*:* $v_1, e_2, \ldots, v_{n-1}, e_n, v$ *and* v_1, e_1, v. $\lightning$

Denote by v *and* v' *the endpoint(s) of the edge* e *we are adding to the graph* G*. Let* $v, e_1, v_1, e_2, \ldots, v_{n-1}, e_n, v'$ *be the path from* v *to* v' *in* G*. By adding the edge* e *and vertex* v *to this path we obtain the cycle:* $v, e_1, v_1, e_2, \ldots, v_{n-1}, e_n, v', e, v$.

$(4) \Rightarrow (1)$: *By our assumption in* (4), G *contains no cycle. It remains to show that* G *is connected, i.e. for any vertices* v *and* v' *there is a path in* G*. Add an edge* e *with endpoints* v *and* v' *to the graph. The resulting new graph will contain a cycle with the edge* e *in it (Why?):* $v', e, v, e_1, v_1, e_2, \ldots, v_{n-1}, e_n, v'$*. Then* $v, e_1, v_1, e_2, \ldots, v_{n-1}, e_n, v'$ *is a path from* v *to* v'.

Trees

Proposition (Vertices of degree 1 in finite acyclic graphs)

If a finite graph G does not contain a cycle and contains at least one edge then there are at least 2 vertices with degree 1 in G.

Proof

Since G is finite, among all paths in G there is at least one path P of maximal length (i.e. a path P such that there is no path longer than P in G). Let P be: $v_0, e_1, v_1, e_2, \dots, v_{n-1}, e_n, v_n$. As G contains at least one edge, the length of P is at least 1, hence $v_0 \neq v_n$.
We show that $deg(v_0) = deg(v_n) = 1$. Proof by contradiction: Suppose that there is an edge $e \neq e_1$ which is incident to v_0. Then the other endpoint v' of e must lie on P, because otherwise $v', e, v_0, e_1, v_1, e_2, \dots, v_{n-1}, e_n, v_n$ would be a path longer than P, a contradiction. Hence $v' = v_k$ for some vertex v_k on P. Then $v_k = v', e, v_0, e_1, v_1, e_2, \dots, v_{k-1}, e_k, v_k$ is a cycle, which is also a contradiction. Therefore there is no edge $e \neq e_1$ incident to v_0 and so $deg(v_0) = 1$. Similarly $deg(v_n) = 1$.

Trees

Theorem (Equivalent characterisations of trees using the number of edges)

For a simple graph G on n vertices ($n \in \mathbb{N}^+$) the following conditions are equivalent:

1. *G is a tree;*
2. *G contains no cycles (i.e. acyclic) and it has $n-1$ edges;*
3. *G is connected and it has $n-1$ edges.*

Proof

If $n = 1$ then the statement is clearly true. (Why?)
$(1) \Rightarrow (2)$: Proof by induction on n: Suppose the statement is true for some $n \in \mathbb{N}^+$. Let G be a tree with $n+1$ vertices. Then G has a vertex v of degree 1. (Why?) Delete vertex v from G. The new graph G' is clearly acyclic. It is also connected: since v can be contained only as an endpoint in any path in G, hence for any vertices v' and v'' in G', the path between v' and v'' in G cannot contain v, and so it is also a path in G'. Therefore G' is connected, hence a tree, and so by our inductive hypothesis it has $n-1$ edges, and so G has n edges.

Trees

Proof (continued)

(2) $\Rightarrow$ (3): *Proof by induction on* n*: Suppose the statement is true for some* $n \in \mathbb{N}^+$*. Let* G *be an acyclic graph with* $n+1$ *vertices and* n *edges. It is sufficient to show that* G *is connected. The graph* G *contains a vertex* v *of degree* 1*. (Why?) Delete* v *from* G*. The resulting graph* G' *is also acyclic and has* n *vertices and* $n-1$ *edges. Hence, by our inductive hypothesis* G' *is connected. Between any vertices* v' *and* v'' *in* G' *there is a path in* G'*, which is also a path in* G*. From any vertex* v' *in* G' *we can obtain a path in* G *to* v *if we consider the path in* G' *from* v' *to the vertex adjacent to* v *in* G *and to this path we add the edge incident to* v *and* v*.*

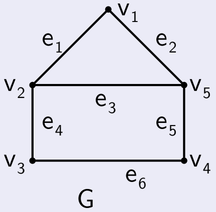
(3) $\Rightarrow$ (1): *Let* G *be a graph satisfying Condition (3). It is sufficient to show that* G *is acyclic. Proof by contradiction: Suppose* G *contains a cycle. Then by deleting any edge of a cycle in* G *we obtain a connected graph. (Why?) Repeat this step, while the graph still has a cycle. The process halts after finite steps (Why?), when the new graph* T *is a tree. If we omitted* $k>0$ *edges during the process, then* T *has* $n-1-k<n-1$ *edges. Because of the implication* (1) $\Rightarrow$ (2), T *has* $n-1$ *edges, a contradiction.* $\lightning$

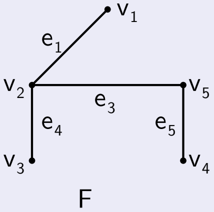
Spanning tree

Definition (spanning tree)

A subgraph T of a graph G is called a spanning tree of G, if T is a tree and T contains all vertices of G.

Example





Spanning tree

Proposition (Spanning tree in finite connected graphs)

Every finite connected graph has a spanning tree.

Proof

Let G be a finite, connected graph. If G contains a cycle then delete an edge from one of the cycles in G. The new graph is still connected. (Why?) Repeat this step until the graph becomes acyclic. This process will terminate in finite number of steps, since the number of edges in the graph decreases in each step. When the process stops, the graph obtained is a spanning tree of G.

Spanning tree

Proposition (Lower bound on the number of cycles)

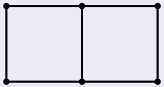
A finite connected graph $G = (\varphi, E, V)$ contains at least $|E| - |V| + 1$ cycles with pairwise different sets of edges.

Proof

Let T be a spanning tree of G. Then T has $|V| - 1$ edges. Denote by E' the set of those edges in G which are not in T. If we add any edge $e \in E'$ to T there will be a cycle K_e in the new graph F' (Because T is a maximally acyclic graph), which is also a cycle in G. The cycle K_e contains the edge e (Why?) and if $e \neq e' \in E'$ then $K_{e'}$ does not contain e. This way we obtain $|E'| = |E| - |V| + 1$ cycles with pairwise different sets of edges.

Comment

The above lower bound does not always give the exact number of cycles in the graph ($3 > 7 - 6 + 1 = 2$).



Forest, spanning forest

Definition (forest, spanning forest)

- An acyclic graph is called a forest.
- A subgraph F of a graph G consisting of one spanning tree in each component of G is called a spanning forest of G.

Proposition (Spanning forest in finite graphs)

Every finite graph has a spanning forest.

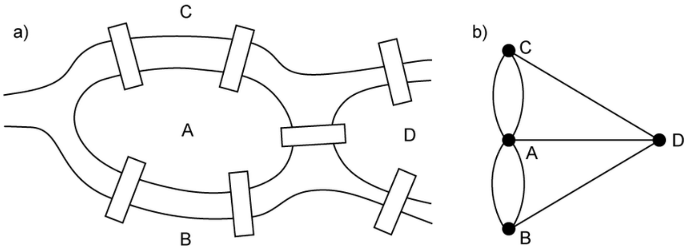
Proposition (Number of edges in a finite forest)

The number of edges in a finite forest F equals the number of vertices in F minus the number of components in F.

Comment

Among all (not necessarily connected) graphs, forests and spanning forests play similar roles to those of trees and spanning trees among connected graphs.

Euler trail



Definition (Euler trail)

A trail that contains all edges of a graph is called an Euler trail.

Comment

Since a trail does not contain repeated edges, an Euler trail contains every edge of the graph exactly once.

Euler trail

Theorem (Existence of a closed Euler trail)

A finite connected graph contains a closed Euler trail if and only if the degree of every vertex in the graph is even.

Proof

$\Rightarrow$: *Let* $v_0, e_1, v_1, e_2, \ldots, v_{n-1}, e_n, v_0$ *be a closed Euler trail in the graph. Following the Euler trail we 'enter' each vertex* v *the same number of times as we 'leave' it. Hence the trail contains an even number of edges incident to any vertex* v *(counting loops twice), hence the degree of every vertex is even.*

Euler trail

Proof

$\Leftarrow$: *The proof is constructive. First consider the closed trail containing no edges (consisting of a single vertex* v*), call it* T*. If the current closed trail* T *does not contain all edges of the graph, then – because the graph is connected – there is a vertex (*v'*) in our closed trail, which has incident edges not included in* T*. Start a new trail* T' *from* v' *leaving* v' *on an edge not used in* T *and proceed always on unused edges. As every vertex has an even number of edges not used in* T*, we can only get stuck when returning to* v'*. Consider the following new trail: on* T *move from* v *to* v'*, then move along* T' *and then from* v' *follow again* T*. This way we obtain a closed trail longer than* T*. Hence, by repeating this expansion step, after a finite number of expansions we obtain a closed Euler trail.*

Hamiltonian path

Definition (Hamiltonian path, Hamiltoninan cycle)

A Hamiltonian path in a graph is a path which contains all vertices of a graph.

Comment

Since a path does not contain repeated vertices, a Hamiltonian path contains each vertex of the graph exactly once.

Definition (Hamiltonian cycle, Hamiltonian graph)

A Hamiltonian cycle in a graph is a cycle that contains all vertices of the graph. A graph is called a Hamiltonian graph if it contains a Hamiltonian cycle.

Theorem (Dirac's theorem)

If in a simple graph $G = (\varphi, E, V)$, $|V| > 2$, and the degree of every vertex is at least $|V|/2$, then the graph contains a Hamiltonian cycle.

Labelled graphs

Definition (labelled graph)

Let $G = (\varphi, E, V)$ be a graph, L_e and L_v be sets, the sets of edge labels and vertex labels, and the maps $\ell_e \colon E \to L_e$ and $\ell_v \colon V \to L_v$ a so called edge labelling and vertex labelling. Then $(\varphi, E, V, \ell_e, L_e, \ell_v, L_v)$ is called a labelled graph.

Definition (edge-labelled and vertex-labelled graphs)

The graph is called edge-labelled or vertex labelled, respectively, if only an edge-labelling, or only a vertex-labelling is given, respectively.

Comment

Labelled graphs are also referred to as coloured graphs.

Labelled graphs

Definition (weighted graph)

In case $L_e = \mathbb{R}$ or $L_v = \mathbb{R}$, we speak about edge-weighting and an edge-weighted graph, or vertex-weighting and a vertex-weighted graph, and from the notation we omit L_e or L_v, respectively.

Definition (weight of a set of edges)

In an edge-weighted graph $G = (\varphi, E, V, w)$ the weight of a set of edges $E' \subseteq E$ is $\sum_{e \in E'} w(e)$.

Kruskal's algorithm

Starting from the empty subgraph (i.e. no edges) of an edge-weighted graph contaning all vertices, in each step add an edge of *minimal weight* to the subgraph such that the new subgraph is still acyclic.

Labelled graphs

Theorem (Kruskal's algorithm)

Kruskal's algorithm determines a spanning forest of minimal weight. In case of connected graphs it finds a spanning tree of minimal weight.

Proof

Not required.

Definition (greedy algorithm)

An algorithm is called a greedy algorithm if in each step it chooses an option from among those that appear to be the most favorable in that given moment.

Comment

Kruskal's algorithm is a greedy algorithm.

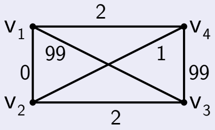
Labelled graphs

Comment

Greedy algorithms are not always optimal.

Example

Find a Hamiltonian cycle of minimal weight in the following graph.


v_1
v_2
v_3
v_4
2
99
1
0
99
2