C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms
* C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph
Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru -
Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru
- Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C.

Graph Algorithms - Lecture 1

Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru -
Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru
- Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C.
Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *
C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms
* C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph

October 4, 2024

Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru -
Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru
- Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C.
Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *
C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms
* C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph
Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru -
Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru
- Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Table of contents

1 Course description
- Course information

2 Graph Theory applications

3 Graph Theory vocabulary
- Graph definition
- Stable sets
- Matchings
- Graph colorings
- Graph isomorphism

4 Exercises for the 2nd seminar (october 7-11 week)

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Course information

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph

Course page:
`https://edu.info.uaic.ro/algoritmica-grafuri`

Objectives:
The lectures will cover basic topics in Algorithmic Graph Theory. Accumulated knowledge will be applied in designing efficient algorithms for combinatorial optimization problems.

Lectures: E. F. Olariu (halfyear A and E), C. Frăsinaru (halfyear B).

- The lecture notes will be posted as .pdf files before each lecture.
- Each of these lectures notes will contain the exercises for the corresponding seminars.

- Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Course information

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph

Seminars: E. F. Olariu, A. C. Frăsinaru, P. Diac, E. Curcă, and A. Ioniţă

- All lectures and seminars will take place onsite.
- Each seminar will contain a short test about past lectures and seminars knowledge.
- During the seminars we will discuss solutions to proposed exercises (which can be found in the corresponding lecture file).
- Each seminar debates three or four exercises (posted in advance in the lecture notes) in order to deepen the subjects introduced in the lectures.
- Students are encouraged to find and present original solutions to these problems.

Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Course information

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms

Scores:

- Seminar: tests (max. $2 \times 11 = 22$p) and activity[a] (max. 20p[b]) - max. 42 points.
- Homeworks: three exercise sheets, 16p each - max. 48 points. (The homeworks can be solved in teams of at most four students, each being leader/responsible for at least one problem from the set.)
- Written final test (compulsory): six exercises (10p each) +10p the basis gives a maximum of 70 points.

From a maximum of 90p for seminar activity and homeworks the threshold for taking the written final test is of 30 points.

From a maximum of 160p the threshold for promovating the course is of 80 points.

[a] Two correct and pertinent answers = 1 point.

[b] Aproximatively.

Course information

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C.

Final grading: the scores will be transformed in grades like follows

- grade 5 for a score in the interval $[80, 85]$;
- grade 6 for a score in the interval $(85, 100]$;
- grade 7 for a score in the interval $(100, 115]$;
- grade 8 for a score in the interval $(115, 130]$;
- grade 9 for a score in the interval $(130, 145]$;
- grade 10 for a score in the interval $(145, 160]$.

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Course information

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms

Grading:

- Seminar. Each test is a question whose not too long answer needs a justification.
- Homeworks. The solutions can be worked in teams of up to four students (each being leader/responsible for at least one problem from the corresponding set). Solutions will be written in LaTeX or Word (Write) and electronically sent (both the source and the corresponding .pdf). 2 points bonus for each homework written in LaTeX. The solution will be checked using an anti-plagiarism tool, the grades of the students which will copy will be penalised (for each copied problem the student will receive -2 points only).
- Written final test (compulsory). The students cannot use course related materials during the test.

- Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Course information

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru

Outline of the course:

- Vocabulary of Graph Theory.
- Path problems: graph traversal, shortest-paths, connectivity.
- Minimum spanning trees: union-find, amortized complexity.
- Matching theory.
- Flows.
- Polynomial time reductions between decision problems on graphs.
- Approaching NP-difficult problems.
- Planar graphs.
- Tree-decomposition.

Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Graph Theory applications

G. Croitoru - Graph Algorithms * G. Croitoru - Graph Algorithms * G. Croitoru - Graph Algorithms

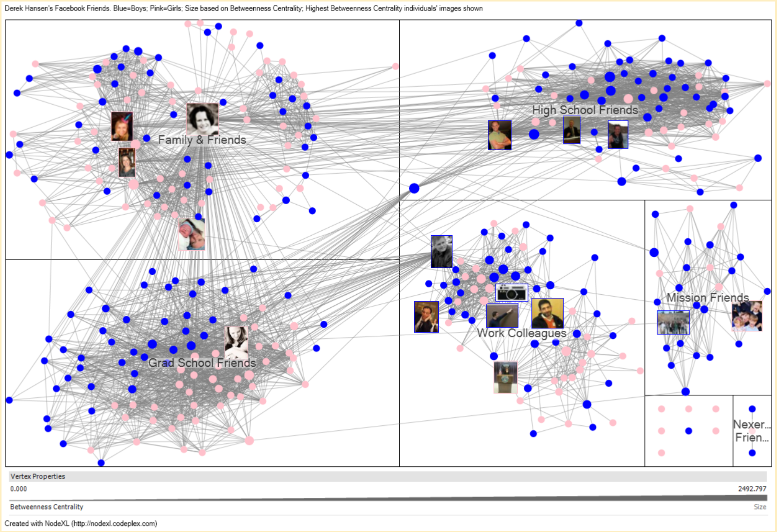

Derek Hansen's Facebook Friends. Blue=Boys; Pink=Girls; Size based on Betweenness Centrality; Highest Betweenness Centrality individuals' images shown
Family & Friends
High School Friends
Work Colleagues
Mission Friends
Grad School Friends
Nexer... Frien...
Vertex Properties
0.000
2492.797
Betweenness Centrality
Size
Created with NodeXL (http://nodexl.codeplex.com)

Figure: Facebook/Twitter

Graph Theory applications

G. Croitoru - Graph Algorithms * G. Croitoru - Graph Algorithms * G. Croitoru - Graph Algorithms

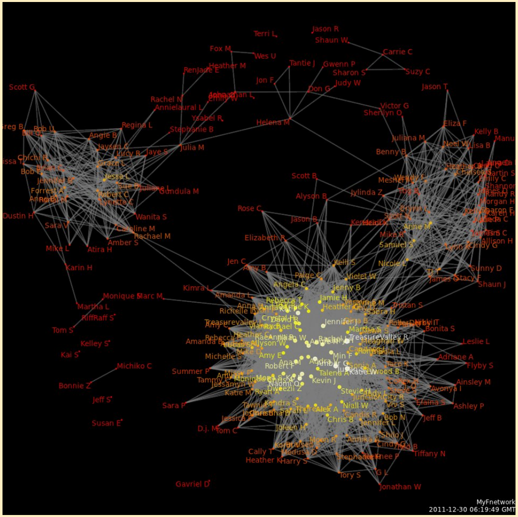


Figure: A Facebook network of a few users

Graph Theory applications

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms (watermark repeated across page)

Graphs are used for analyzing social/news networks (like Facebook or Twitter) in order to find characteristics like:

- the level of connectivity, the density;
- the influence of users within the network (centrality, the social networking potential);
- the level of segmentation: measuring the clusterization;
- robustness or structural stability of the network.

The network analysis is then used for

- data mining and aggregation, marketing;
- behavior analysis and network prediction;
- intelligence and security analysis (surveillance of terrorism and organized crime) etc.

Graph Theory applications

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru -

Apart from social network study which is a hot topic nowadays there are a lot of other applications for graph algorithms:

- Minimum spanning tree: for efficiently connecting communicating points (e. g. in IT&C);
- Eulerian paths and cycles: The chinese postman problem - find a shortest closed path/cycle that visits once every edge of a graph (for street cleaning, mail/services/packages delivery, garbage collection etc.);
- Hamiltonian paths and cycles: efficiently visiting once a number of points (in a city, a country etc); Travelling salesman problem, Vehicle routing problem;

Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Graph Theory applications

- Graph colorings: coloring maps (faces of planar graphs), lectures/seminars/timetabling, exam scheduling for university departments, mobile radio frequency allocation, memory register allocation.

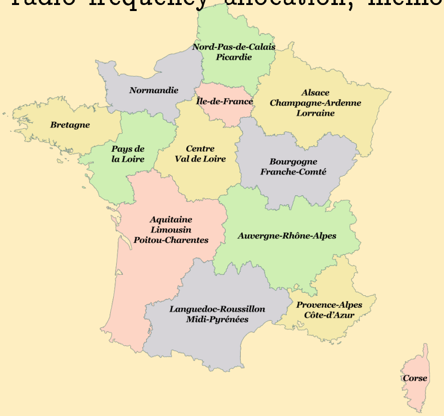
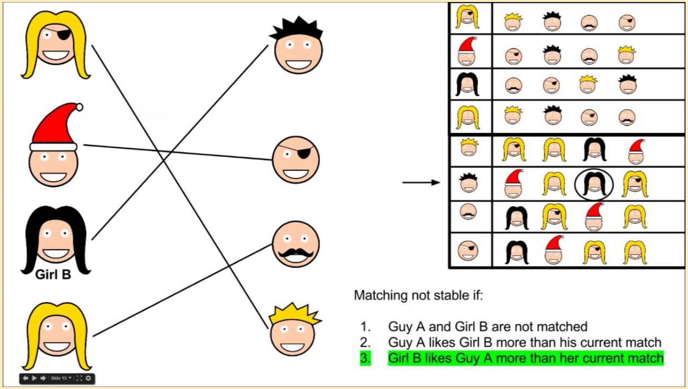


Figure: France regions from 2016

Graph Theory applications

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms

- Matchings: assignment problems, in computational chemistry and mathematical chemistry studies on organic materials.



Matching not stable if:

1. Guy A and Girl B are not matched
2. Guy A likes Girl B more than his current match
3. Girl B likes Guy A more than her current match

Figure: Gale Shapley algorithm

Graph Theory applications

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C.

- Stable matchings: assignment resident physicians to hospitals, the admission procedure of higher education institutions, identifying optimal assignment of kidney (in transplantations), student-project allocation problem etc.
- Maximum flow: find the charging level in transportation networks for improving the traffic parameters, image reconstruction from X-Rays projections (in tomography), scheduling on parallel processors.

Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Graph Theory applications

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph

Applications in Computer Science

- In database management, graph database uses graph-structure data for storing and querying.
- Graph rewriting systems are used in software verification.
- Quantum computation.
- Modeling the web documents as graphs and clustering of web documents.
- Approximation of data and data compression.
- Modeling the sensor network as graphs (using Voronoi diagrams).
- Graph Neural Networks in AI.
- And a lot more.

Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Graph definition

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C.

Notations

For a given finite set X:

- $|X| = card(X) \in \mathbb{N}$ is the cardinality of X;
- If $|X| = k$, then X is a k-set;
- $2^X = \mathcal{P}(X)$ is the power set of X: $2^X = \{Y : Y \subseteq X\}$, $|2^X| = 2^{|X|}$;
- $\binom{X}{k} = \mathcal{P}_k(X) = \{Y : Y \subseteq X, |Y| = k\}$, $\left|\binom{X}{k}\right| = \binom{|X|}{k}$.

Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Graph definition

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C.

Definition 1

A graph is a pair $G = (V, E)$, where:

- $V = V(G)$ *is a finite, non-empty set; is the set of vertices (nodes) of* G;
- $E = E(G)$ *is a subset of* $\binom{V(G)}{2}$; *is the set of edges of* G.

$|G| = |V(G)|$ *is the order of graph* G, *and* $|E(G)|$ *is the size of* G.

Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Graph definition

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Definition 2

If $G = (V, E)$ and $e = uv = vu = \{u, v\} \in E$ is an edge of G, then we say that

- *e connects (or links) vertices u and v; that*
- *vertices u and v are adjacent or u and v are neighbors;*
- *e is incident with u and v;*
- *u and v are the endpoints (extremities) of e.*

Neighborhood of vertex u is $N_G(u) = \{v \in V(G) : uv \in E(G)\}$.

Two edges e and f are adjacent if they share a common endpoint: $|e \cap f| = 1$.

Graph definition

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph

Let $G = (V, E)$ be a graph and $v \in V$.

- the degree of vertex v: $d_G(v) =$ number of edges incident to v.
- v is an isolated vertex if $d_G(v) = 0$; v is called pendant (or leaf) if $d_G(v) = 1$.
- an useful property:

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

Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru -

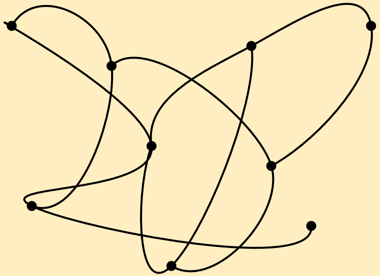
A graph can be represented in plane as a figure consisting of a set of nodes (small geometric forms: points, circles, squares etc) corresponding to its vertices and curves which connect the vertices corresponding to the edges from the graph.

- Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Graph definition

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru

An example:

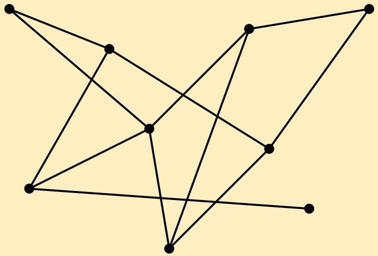


C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Graph definition

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C.

The same graph again:



C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

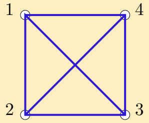
Graph definition

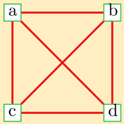
C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru -

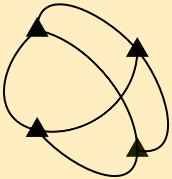
We can add labels (names, numbers etc) and colors to vertices and edges obtaining better visual representations.

Below there are three visual representations of the same graph:

$$G = (\{1, 2, 3, 4\}, \{12, 13, 14, 23, 24, 34\})$$







Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Stable sets

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru -

Definition 3

Stable set (or independent set of vertices) in $G = (V, E)$: $S \subseteq V$ such that $\binom{S}{2} \cap E = \varnothing$.

- Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C.

In other words $S \subseteq V$ is a stable set of G if there is no edge between its vertices.

Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru -

Notation

The maximum cardinality of a stable set of G is the stability number (or independence number) of G and is denoted by $\alpha(G)$.

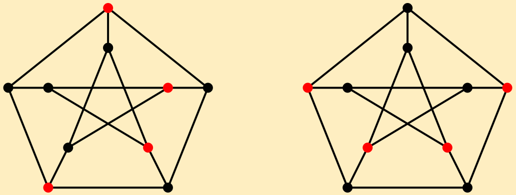
Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Stable sets

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru

Example

In the following graph (Petersen's) we have two different stable sets of maximum (why?) cardinality:



* C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Stable sets

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru -

Stable sets: Optimization problem

P1 Input: G graph.
Output: $\alpha(G)$ and a witness stable set S with $|S| = \alpha(G)$.

- Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C.

Stable sets: Decision problem

SM Instance: G graph, $k \in \mathbb{N}$.
Question: Is there a stable set S in G, such that $|S| \geqslant k$?

NP-complete (Karp, 1972).

Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Matchings

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Definition 4

Matching (independent set of edges) *in* $G = (V, E)$: $M \subseteq E$ *such that for all* $e, f \in M$, *if* $e \neq f$, *then* $e \cap f = \varnothing$.

In more words, $M \subseteq E$ is a matching if each pair of its edges share no endpoint.

Notation

The maximum cardinality of a matching in G is called the matching number (edge-independence number) of G and is denoted by $\nu(G)$.

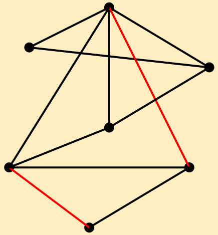
Matchings

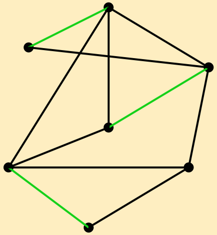
C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru -

Example

For the following graph two matchings are depicted with red and green - the second being of maximum (why?) cardinality:

Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru





Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Matchings

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru -

Maximum matching: Optimization problem

P2 Input: G graph.
Output: $\nu(G)$ and a witness matching M with $|M| = \nu(G)$.

Edmonds (1965) showed that $P2 \in$ P.

Note

The problems $P1$ and $P2$ are similar: in both we are required to find a member of maximum cardinality of a family of sets concerning a given graph. What makes the difference?

Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Graph colorings

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph

Definition 5

For $p \in \mathbb{N}^*$, a p-coloring of the graph $G = (V, E)$ is a map $c : V \to \{1, \dots, p\}$ such that $c(u) \neq c(v)$ for each $uv \in E$.

It is worthnoting that the set of all vertices having the same given color is a stable set (it is called also a coloring class). Since adjacent vertices have different colors, a p-coloring is a partition of V with at most p stable sets (or coloring classes).

Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru

Notation

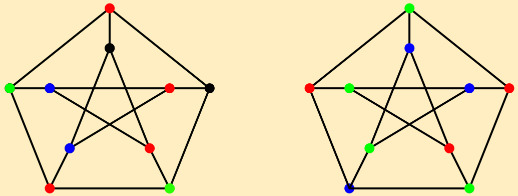
The chromatic number of the graph G is the least value of p such that G has a p-coloring. This parameter is denoted by $\chi(G)$. ($\chi(G) \leqslant |G|$ - why?)

Graph colorings

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C.

Example

For the following graph we depicted two colorings ($\chi(G) = 3!$)



Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Graph colorings

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru -

Vertex coloring: Optimization problem

P3	Input:	G graph.
	Ouput:	$\chi(G)$ and a witness $\chi(G)$-coloring.

- Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C.

Vertex coloring: Decision problem

COL	Instance:	G graph, $p \in \mathbb{N}$.
	Question:	Is there a p-coloring of G?

NP-complete for $p \geqslant 3$ (Karp, 1972).

Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Graph colorings

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph

Definition 6

For $p \in \mathbb{N}^*$, a p-edge coloring of the graph $G = (V, E)$ is a map $c : E \to \{1, \dots, p\}$ such that $c(e) \neq c(f)$ for all $e, f \in E$ with $|e \cap f| = 1$.

We can note that in a p-edge coloring a set of edges with the same color is a matching. Hence, a p-edge coloring is a partition of E with at most p matchings.

Notation

Chromatic index of the graph G: the least value of p such that G has a p-edge coloring. This parameter is denoted by $\chi'(G)$. ($\chi'(G) \leqslant |E(G)|$ - why?)

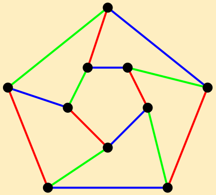
- Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Graph colorings

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C.

Example

For the following graph we depicted an edge coloring ($\chi'(G) = 3!$)



Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Graph colorings

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru -

Edge coloring: Optimization problem

P3 Input: G graph.
Ouput: $\chi'(G)$ and a witness $\chi'(G)$-coloring.

- Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C.

Edge coloring: Decision problem

COL Instance: G graph, $p \in \mathbb{N}$.
Question: Is there a p-edge coloring of G?

NP-complete for $p \geqslant 3$ (Holyer, 1984).

Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Graph isomorphism

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph

Definition 7

Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic *if there is a bijection $\varphi : V_1 \to V_2$ such that for every two vertices $u_1, v_1 \in V_1$, u_1 and v_1 are adjacent in G_1 (i. e., $u_1v_1 \in E_1$) if and only if $\varphi(u_1)$ and $\varphi(v_1)$ are adjacent in G_2 (i. e., $\varphi(u_1)\varphi(v_1) \in E_2$).*

- Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

In other words two graphs are isomorphic if there is a bijection between their sets of vertices which induces a bijection between their sets of edges.

Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Notation

$G_1 \cong G_2$.

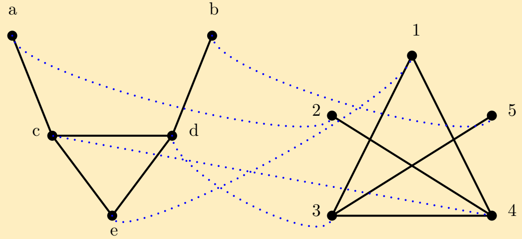
Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Graph isomorphism

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru

Example

The graphs from below are isomorphic.



C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Graph isomorphism

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C.

Isomorphism testing: Decision problem

ISO	Input:	graphs G_1 and G_2.
	Ouput:	are G_1 and G_2 isomorphic?

Neither known to be in P nor NP-complete.

There is a quasipolynomial time algorithm (i. e., running in $2^{\mathcal{O}((\log n)^c)}$ for some $c > 0$, Babai, 2015).

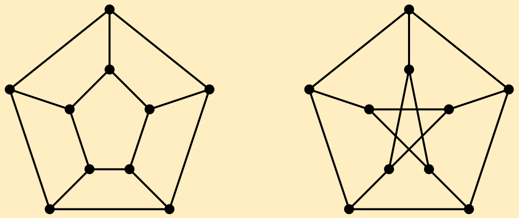
Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Graph isomorphism

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru

Example

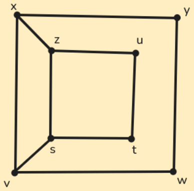
The following two graphs have the same order, size and degree sequence, but they are not isomorphic (why?).

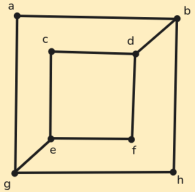


* C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Exercises for the 2nd seminar

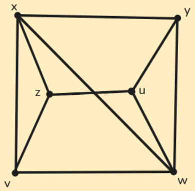
Exercise 1. The following graphs are isomorphic?

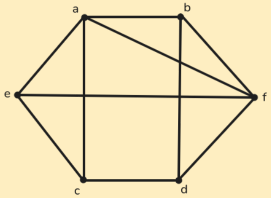




Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *
C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms

Exercise 2. The following graphs are isomorphic?

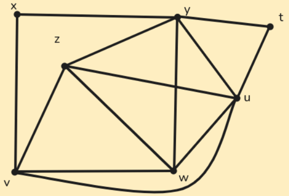


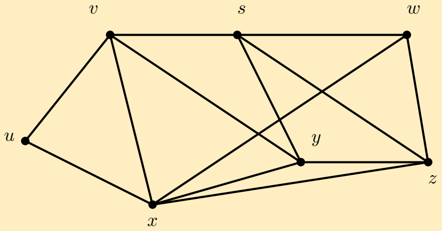


Exercises for the 2nd seminar

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C.

Exerciţiul 3. Find the chromatic number of the following graphs:



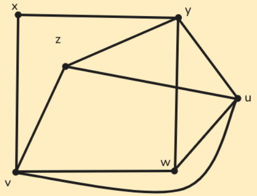


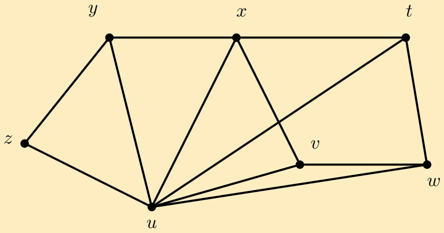
Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Exercises for the 2nd seminar

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph

Exercițiul 4. Find the stability number of the following graphs





Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru -

Exercise 5. Let $P(n) =$" In any graph with at least n vertices there are three distinct vertices which are pairwise adjacent or pairwise non-adjacent." Prove that 6 is the least value of $n \in \mathbb{N}, n \geqslant 3$ for which $P(n)$ is true.

Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Exercises for the 2nd seminar

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C.

Exercise 6.

(a) Is there a graph with degrees $2, 3, 3, 3, 4, 4, 5, 6, 6$?

(b) Is there a bipartite graph with degrees $3, 3, 3, 3, 3, 5, 6, 6, 6, 6, 6, 6, 6, 6$?

(c) Is there a graph with degrees $1, 1, 3, 3, 3, 3, 5, 6, 8, 9$?

(d) Is there a graph with degrees $1, 2, 2, 3, 3, 3, 6$?

(e) Is there a graph with degrees $2, 2, 3, 3, 3, 3, 6, 6$?

Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Exercises for the 2nd seminar

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru -

Exercise 7. Two students, L(azy) and T(hinky), must find a particular path between two given vertices in a sparse graph G: $|E(G)| = O(|V(G)|)$. L judge that, since the graph is sparse, the number of paths between the two vertices is small, and a lazy solution is to generate (backtracking) all these paths and than retain the wanted one. T does not agree and gives the following example: let $H_n = K_2 \times P_n$ $(n \geqslant 1)$; add to H_n two new vertices x and y each joined by two edges with one of the two pairs of adjacent vertices in H_n of degree 2.
The obtained graph, G_n, is sparse but the number of paths between x and y is big. Explain to L this example by drawing G_n, showing its sparsity, and finding the number of paths between x and y.

Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Exercises for the 2nd seminar

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru

Exercise 8. An examination session must be scheduled based on the following input specifications: the set of exams is known; each student sends the list of exams to which (s)he is registered; each exam take place with all students registered to it (written exam); each student can participate to at most one exam in the same day.
Construct a graph in order to answer the following questions (by determining appropriate parameters):

(a) What is the maximum number of daily exams?

(b) What is the minimum number of days for the examination session?

* C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Exercises for the 2nd seminar

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru -

Exercise 8'. (exercise 8 cont'd) A smart and skilled programmer is wondering if the two NP-hard problems in the above exercise can be solved in polynomial time since the graph constructed seems to belong to a very special class of graphs.

(a) Prove that for any graph, G, there is an input for the scheduling problem above such that the graph constructed is G.

The programmer suggests the following "greedy approach" to solve the second question in exercise 8: starting with day 1, a maximum number of exams are scheduled (from the set of exams not scheduled) daily, until all exams are scheduled.

b) Show that this greedy strategy can fail, by giving a counterexample.

Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Exercises for the 2nd seminar

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru -

Exercise 9. A compiler optimization is the register allocation technique: the most used variables are kept in the fast processor registers in order to have them there at the moment when the compiler needs them (for certain operations made by CPU).
Design a graph that - by using its appropiate parameters - will answer to the following questions:

(a) What is the maximum number of variables which are not needed at the same time?

(b) What is the minimum number of registries needed?

Hint: We have two kind of objects at hand: variables (with their values) and CPU operations (or operators) which use one or more variables.

Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Exercises for the 2nd seminar

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru -

Exercise 9'. (exercise 9 cont'd) A student is wondering if the above questions (which correspond to NP-hard problems) can be answered by polynomial time algorithms since the graph constructed seems to belong to a very special class of graphs.

(a) Prove that for any graph G there is an input for the registry allocation problem such that the resulted graph is G.

The student suggests the following "greedy approach" to answer the second question in exercise 9: starting with the first step, allocate as many as possible variables to one new free register in each step, until there are no more variables.

(b) Show that this greedy approach fails, by giving a counterexample.

Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *

Exercises for the 2nd seminar

C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C.

Exercise 10. G is called *self-complementary* graph if G and its complement, $\overline{G}$, are isomorphic ($G \cong \overline{G}$).

(a) Show that a self-complementary graph is connected and $|G| \equiv 0$ or 1 *mod* 4.

(b) Find all self-complementary graphs with at most 7 vertices.

(c) Show that, for every graph G, it exists an self-complementary graph H, such that G is an induced subgraph in H,

Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms * C. Croitoru - Graph Algorithms *