|  |  |
| --- | --- |
| The title of the course | **Algorithmic Graph Theory** |
| Faculty | [Faculty of Mechanical Engineering and Computer Science](http://eng.ath.bielsko.pl/index.php/faculties/gerg) |
| The level of studies | Engineer (BSc) |
| Semester | Winter/summer |
| Classes conducted for Polish students. Erasmus students can join them | YES |
| The form of classes and number of hours | Lectures/laboratories  Regular Lectures or Tutorial system |
| Language of instruction | English |
| The number of ECTS | 3 |
| Teacher | Assoc. Professor Stanisław Zawiślak |
| The aims of the course  (maximum 500 characters) | The course covers versatile algorithms for several graph theory problems i.e.: generation (of families of trees and graph generation in general), graph drawing, search, discrete optimization, enumeration etc. Therefore the aims are: to learn the algorithms, to apply different data structures, to learn how graphs can be presented visually. The comparison of different algorithms for the same problem is discussed e.g. spanning tree problem – the algorithms of Prim, Kruskal and Boruvka are taken into account. Student task is to write own computer program performing the chosen graph theory algorithm – so the aim is to turn an algorithm into own computer program. Practical algorithms are considered e.g. the shortest path, the Hamiltonian path and Euler cycle. Some application of utilization of graphs in modelling of mechanical systems is additionally discussed. |
| The content of the course: main topics and key ideas | Graph generation – cliques, graphs where number of edges is assumed, graphs where existence of an edge is defined by a probability ‘p’ and other schemes of graph generation.  Graph drawing – the methods of drawing graphs are discussed: random distribution of vertices, even distribution on a circle or on a net.  Data structures for encoding graphs: matrices and lists.  Depth and breadth graph search – connectivity detection.  Spanning tree number calculation and tree generation – codes of trees, data structures of trees.  Graph three isomorphism checking for some cliques – finding a abstraction classes of isomorphic trees.  Spanning tree algorithms – comparison.  Different algorithms for shortest path – Ford, Danzig, Kallaba.  Max flow – minimum cut algorithm.  Visualisation of algorithms is crucial for didactic purposes. |
| Didactics methods | Lectures, usage of computer programs,  Preparation/adaptation of algorithms and writing own computer programs |
| Course requirements | Presentation/attendance/computer program with written description – it means the report |
| Literature (basic and supplementary) | 1] Andrafai B.: Introductory graph theory, Budapest, Akademiai Kiado, 1977.  2] Zawislak S.: The graph-based methodology as an artificial intelligence aid for mechanical engineering design, ATH, Bielsko-Biala 2010.  3] Wilson R.J.: Introduction to graph theory, London, 1985  4] Even S.: Graph algorithms, Computer Science Press, 1979.  5] West D.B.: Introduction to graph theory, Prentice Hall, New York, 1996.  A1] Jungnikiel: Graphs, Networks and Algorithms, Springer, 1999.  A2] Kozo Sugiyama: Graph drawing and applications for software and knowledge engineering, World Scientific, 2002.  A3] Tero Harju: Lecture notes on graph theory, University of Turku, available as pdf file.  A4] Tsai L.-W.: Mechanism design. Enumeration of kinematic structures according to function, CRC Press Boca Ranton, Florida 2001.  A5] Kubale M.: Graph colorings, American Mathematical Society, 2004 . |
| The effects of the education   * knowledge * skills * social competences | Knowledge:  IDI.25\_W01: student has a basic knowledge within the range of building graph-theory algorithms and applying them in own software;  IDI.25\_W02: She/He has a basic knowledge on modelling of engineering problems by means of graphs  Skills:  IDI.25\_U01: She/He can write a computer program performing chosen graph algorithm  IDI.25\_U02: She/He can analyse the complexity and run of the computer program prepared by herself/himself  Social competence (depending on number of students):  IDI.25\_K01: She/He works in a team, sharing programming tasks in such a way that the program unifies all parts of team members  IDI.25\_K02: She/He recognized a need of self-education and is able to do it a team; especially sharing the task of internet browsing. |