I019	FIN, IPM - elective –Year 1 MR - obligatory – Semester 1	**Algorithm Complexity**	L+P+S 2+2+0	ECTS 5

Course objectives. The main course objectives are to teach students advanced concepts in algorithms and randomized algorithms. Moreover, one of the main goals will be to instruct students how to improve the ability of programming (C++, Java) on more complex data structures and algorithms.

Course prerequisites. Introduction to Programming, Introduction to Data Structures and Algorithms

Syllabus.

1. Introduction. Different kinds of algorithms and the standard techniques for solving them.
2. Lower bound for sorting with comparisons. Sorting in linear time (RadixSort, CountingSort, BucketSort).
3. QuickSort algorithm, advantages and disadvantages. Randomized QuickSort algorithm and its expected running time. Analysis of the expected running time of QuickSort algorithm in asymptotic notation.
4. Linear time randomized algorithm for median computation. Derandomized version of the algorithm. Combining algorithm for median with QuickSort algorithm.
5. Hash tables. Universal hashing.
6. Amortized analysis. Dynamic tables and comparison with C++ STL "vector" class.
7. Usefulness of the potential function in complexity analysis. Competitive analysis. On-line algorithms. K-competitive algorithms. Self-organizing lists.
8. Approximation algorithms. Vertex covering problem. Traveling salesman problem. Set-cover problem. Subset-sum problem. A randomized approximation algorithm for the MAX-3-CNF problem.

Expected learning outcomes.

After completing the course, students are expected to:

- demonstrate the knowledge and intelligence as the basis for the original work and development of ideas;
- apply their knowledge, understanding and ability to problem solving in a wider context in the area of algorithms;
- be capable of integrating new knowledge in the area of algorithm complexity;
- be able to communicate their conclusions and supporting arguments to both experts and non-experts;
- possess the learning ability for continuing education and lifelong learning in this area.

Teaching methods and student assessment. During lectures students will be introduced to the theory and basic problems as well as with the ways of solving the. Exercises will be held in specialized computer-based laboratories where students will learn how to analyze and create algorithms for different problems. The final exam will be held after completion of lectures and exercises and it will contain a practical and an oral part. Successful participation in mid-term exams (or homework) replaces obligatory participation in the practical part of the exam. Students can influence their final grade if they actively participate in homework assignments during the semester.

Can the course be taught in English: Yes.

Basic literature:

1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms. 2nd Edition, MIT Press, 2001.

Recommended literature:

1. V. V. Vazirani, Approximation Algorithms. Springer, 2003.
2. K. Melhorn, Efficient data structures and algorithms, 3rd Edition, Springer, 2003.
3. D. Knuth, The Art of Computer Programming, Vol. 1, Fundamental Algorithms, Addison-Wesley, Reading, MA, USA, 1997.
4. M. T. Goodrich, R. Tamassia, D. Mount, Data Structures and Algorithms in C++, John Wiley and Sons, 2011.