**Table 5.2**Subject Specification

| | | | |
|---|---|---|---|
| **Study program:Master of Science degree in Electrical Engineering and Computer Science** | | | |
| **Name of the subject: Analysis and Design of Algorithms** | | | |
| **Teacher: Vladimir M. Milovanović** | | | |
| **Status of the subject:** elective | | | |
| **Number of ECTScredits:** 6 | | | |
| **Conditions:** none | | | |
| **Subject goal**<br>Analysis of asymptotic complexity of algorithms, writing rigorous proofs of correctness, introduction to the most commonly used algorithms and data structures, application of design techniques and analysis methods, as well as synthesis of time and space efficient algorithms in engineering practice. | | | |
| **Outcome of the subject**<br>Mastering the necessary knowledge and skills for rigorous proof of correctness, as well as for asymptotic analysis of algorithm complexity. Introduction to and application of recursive strategies based on decomposition, greedy and probabilistic algorithms, dynamic programming, as well as cost amortization techniques. Student well prepared for a typical programming interview. | | | |
| **Subject content**<br>*Theory*<br><br>Short overview on fundamental linear and nonlinear data structures and algorithms which operate on them. Arrays. Lists. Stacks. Queues. Trees: binary trees, minimum path length, traversal. Graphs. Representation. breadth-first search, depth-first search, spanning and minimum spanning trees, shortest path and reachability minimum cut and maximum flow, topological sort and critical path. Search. Fundamental methods and improvements. Binary search tree, AVL tree, optimum binary search trees. B, B* and B+ trees. Tries. Hashing. Hash functions, collision avoidance. Sorting. Comparison sorts: insertion sort, selection sort, linear-time sorting. Order statistics. Heaps. Loglinear sorting algorithms. Lower-bound of sorting algorithms. Algorithm analysis. Examples of algorithm correctness proves. Asymptotic analysis of worst or average case. Time and memory complexity. Finite sums, recurrent relations, fundamental theorems. Dynamic programming. Algorithm strategies. Brute-force algorithms. Greedy algorithms. Back-track algorithms. Divide and conquer algorithms. Examples of numeric algorithms. Tail recursion. Randomized algorithms.<br><br>*Practical learning* | | | |
| **Literature**<br>1. T. Cormen, C. Leiserson, R. Rivest, C. Stein, "Introduction to Algorithms", 3rd edition, MIT Press, 2009.<br>2. R. Sedgewick, K. Wayne, “Algorithms”, 4th edition, Addison-Wesley Professional, 2011.<br>3. D.Knuth, "The Art of Computer Programming, Volumes 1-4A Boxed Set",3rd edition, Addison-Wesley, 2011.<br>4. G. McDowell, "Cracking the Coding Interview", 6th edition, CareerCup, 2015. | | | |
| **Number of active teaching classes** | **Theoretical teaching:** 30 | **Practical teaching:** 30 | |
| **Method of carrying out the teaching**<br><br>Lectures, office hours and independent research work under the teacher’s mentorship. | | | |
| **Evaluation of knowledge (maximum number of points 100)** | | | |
| **Pre-exam obligations:** | 70 points | **Final exam:** | 30 points |