Program:

1. Review of methods of designing efficient algorithms: divide and conquer, programming
   dynamic, greedy method. (4 hours).
2. The computational complexity of the algorithm (pessimistic, expected, depreciated). Examples
   cost analysis. (2 hours.)
3. Sort: Heapsort and Quicksort. Decision tree model and a lower limit to the problem
   sorting. Sorting in linear time: Countsort, Radixsort, Bucketsort. (6 hours).
4. Selection: Hoarea algorithms and magical Friday. (2 hours.)
5. Priority Queues: mounds binary, binomial and Fibonacci. Applications in the problem
   shortest paths and minimum spanning tree. (4 hours).
6. Merge. Trees tournament. External sorting. (2 hours.)
7. Search and the problem of the dictionary. Binary search tree, balanced tree
   binary search (AVL 2-3-4-trees, red-black tree). optimal tree binary search. Hashing. Search position. (8 hours).
8. External Search - B-tree. (2 hours.)
9. The problem of aggregation of disjoint sets and their use. (4 hours).
10. Graph algorithms: flows in networks, associations. (4 hours).
11. Algorithms on texts. Search pattern. Trees suffix. (4 hours).
12. Computational Geometry. Location point. Convex hull. Sweeping technique. (4 hours).
13. Algorithms algebraic number theory. FFT. Fast multiplication of numbers and polynomials. (4 hours).
14. NP-completeness. Approximation algorithms for computationally difficult problems. heuristics
    for difficult problems (genetic algorithms, simulated annealing). (4 hours).
15. Models of parallel computation: PRAM array processors, hiperkostka. Parallel algorithms.
    NC and class P-complete problems. (2 hours.)
16. Special computation models: network comparators, logic circuits. (2 hours.)
17. randomization algorithms. Examples in the fields of data structures, computational geometry,
    Graph algorithms, parallel algorithms. (2 hours.)

Requirements: Programming and Discrete Mathematics