

Vistula, IT Faculty, 2014

# Algorithms and Complexity

Dmitry A. Zaitsev

<http://daze.ho.ua>

## *Basic topics*

1. Intuitive concept of an algorithm
2. Algorithm of Euclid
3. Features of algorithms – generality
4. Features of algorithms – unambiguity
5. Features of algorithms – constructivism
6. Features of algorithms – elementarism of steps
7. Hilbert list – 10<sup>th</sup> problem
8. Known models of computations
9. Thesis of Church
10. Undecidable problems
11. Intractable problems
12. Tractable problems
13. Correctness of algorithms
14. Forms of algorithms representation
15. Flow chart as a form of algorithms representation
16. Paths in a triangle of numbers – greedy approach
17. Paths in a triangle of numbers – exhaustive search
18. Paths in a triangle of numbers – dynamic programming
19. Components of Turing machine
20. Rules of Turing machine work
21. Representing TM – state diagram
22. Representing TM – tabular form
23. Representing TM – set of instructions
24. Formal definition of Turing machine
25. Variants of TM definition
26. Multitape TM
27. Weak TM
28. Programming in TM: sequence, branching, loops
29. Numerical encoding of TM
30. Concept of a universal TM
31. Problem of TM self-applicability
32. The problem of TM self-applicability is undecidable
33. Technique of reduction to prove undecidability
34. Undecidability of TM halting problem
35. Enumeration of TM
36. TM and languages
37. Post correspondence problem
38. Problem of matrices representation
39. What to do with undecidable problems
40. Time complexity of TM

41. Space complexity of TM
42. Complexity in the worst (best) cases and on average
43. Asymptotic estimations of complexity
44. Big, small “o” and omega notation
45. Basic rules to treat big-o expressions
46. Time and space complexity of a program
47. Linear and logarithmic scale for complexity evaluation
48. Polynomial complexity
49. Exponential complexity
50. Growth of numbers during computation
51. Mapping one dimension array into memory
52. Mapping matrix into memory
53. Static and dynamic memory allocation
54. Representing tables
55. Sequential search in tables
56. Representing sparse matrices
57. Representing sets and bags
58. Abstract lists
59. One link lists
60. Representing stacks via one link lists
61. Representing queues via one link lists
62. Two link and cyclic lists
63. Undirected graphs
64. Directed graphs
65. Trees and forests
66. Breadth-first search
67. Depth-first search
68. Minimum cost spanning tree
69. Kruskal’s algorithm
70. Algorithm of Deijkstra for minimal paths
71. Travelling salesman problem
72. Concept of recursion and its implementation
73. Computing factorial via recursion – tracing of calls
74. Computing Fibonacci numbers via recursion
75. Recursive solution of Hanoi tower problem
76. Recursive algorithms for lists
77. Recursive algorithms for binary trees
78. Sorting via sequential minimums
79. Sorting via merging
80. Quick sorting via partitioning
81. Bucket sorting of integers
82. String matching algorithms
83. Dichotomy in a sorted table
84. Binary tree search
85. Hash tables with overflow chains
86. Hash tables with open hashing
87. Search using numerical trees
88. Nondeterministic Turing machine
89. Computation of nondeterministic TM
90. P and NP problems
91. Simulating nondeterministic TM on deterministic TM
92. Whether  $P=NP$ ?

93. Concept of NP-completeness
94. Polynomial reduction of problems
95. Polynomial reduction 3SAT to clique
96. NP-completeness of Boolean expression satisfiability
97. Sketch of Cook theorem proof
98. Examples of NP-complete problems
99. Classes of space complexity
100. What to do with NP-complete problems
101. Divide-and-conquer approach
102. Multiplication of integers – divide and conquer
103. Sorting via merging – divide and conquer
104. Branch and bound approach
105. Principles of dynamic programming
106. Computing Fibonacci numbers via DP
107. Multiplication of matrices via DP
108. Solving knapsack problem via DP
109. Paradigms of programming

