Federal State Autonomous Educational Institution of Higher Education "Moscow Institute of Physics and Technology (National Research University)"

APPROVED
Head of the Phystech School of Applied Mathematics and Informatics
I.V. Raygorodskaya

Programme for the final state attestation (defence of the graduation thesis)
Preparation for and Taking State Examination in Informatics and Discrete Mathematics/Подготовка к сдаче и сдача государственного экзамена по информатике и дискретной математике

by direction (speciality):	Applied Mathematics and Informatics
orientation (profile):	Computer Science/Информатика
	Chair of Algorithms and Programming Technologies
course:	4
qualification:	Bachelor

semester: 7 (Fall)

Программу составили:

V.V. Yakovlev, candidate of physics and mathematical sciences, head of chair
K.Y. Voytikov, candidate of technical sciences, associate professor, associate professor

The programme was discussed at a meeting Chair of Algorithms and Programming Technologies 04.06.2020

1. Goals and objectives

Goals

The purpose of the state exam in mathematics is to establish the level of training of the student in mathematical disciplines and the compliance of the results of mastering the educational program with the requirements of the educational standard in the direction of preparation.

Objectives

- Assessment of the degree of mastering by students of the theoretical provisions of the main disciplines that form special knowledge in the framework of mastering the educational program;
- assessment of the ability to apply the knowledge gained to solve specific problems;
- assessment of the relevance of the knowledge gained by students and their compliance with the requirements of potential employers.

2. List of competences, the level of which is assessed in the state examination

Code and name of competence	Indicators of competence achievement
Gen.Pro.C-1 Apply fundamental knowledge of physics, mathematics, and/or natural sciences in professional settings	Gen.Pro.C-1.2 Build mathematical models, make quantitative measurements and estimates
Pro.C-1 Assign, formalize, and solve tasks, develop and research mathematical models of studied phenomena and processes, systematically analyze scientific problems, obtain new scientific outcomes	Pro.C-1.2 Make hypotheses, build mathematical models of the studied phenomena and processes, evaluate the quality of the developed model
Pro.C-2 Conduct scientific research and testing independently or as a member (leader) of a small research team	Pro.C-2.3 Present research results through scientific publications and participation in conferences

3. List of sample questions for the state examination

1. Set and subset concepts. Operations on sets, identities. Mappings and matching. Comparison of sets by cardinality. Cantor – Bernstein theorem. Countable sets and their properties. Cantor's theorem.
2. Boolean functions and propositional formulas. Conjunctive and disjunctive normal forms. Tautology. Propositional calculus: axioms, rules of inference, definition of derivability, examples of inferences. The correctness of the calculus of statements. Lemma on deduction. Completeness of the propositional calculus: formulation and idea of the proof.
3. First-order languages: signatures, terms, rules for constructing formulas. Interpretations, evaluations, determination of the truth of the formula. Expressibility of predicates: definition, examples, proof of inexpressibility using automorphisms.
4. Generally valid formulas of the first order. Calculus of predicates: formulas and rules of inference. Examples of conclusions: replacing the order of quantifiers and the interaction of quantifiers and logical operations. Correctness of the predicate calculus: formulation and idea of the proof. Gödel's theorem on the

completeness of the predicate calculus: different formulations and general scheme of the proof.
5. Turing machines. Computable functions. Solvable and enumerable sets, their properties. Undecidability of self-applicability and stopping problems. Rice – Uspensky theorem (b / d). Kleene fixed point theorem (b / d). The existence of a program that prints its own text.
6. Formal arithmetic. Examples of conclusions in Peano's axiomatics. Modeling Turing machines in formal arithmetic (b / d). Gödel's Incompleteness Theorem: Statement and Idea of Proof.

7. Lambda calculus. Lambda terms and combinators. Conversions: alpha conversion and beta reduction. Normal form. Church – Rosser theorem (b / d). Church Numerals. Combinators representing addition and multiplication. Representation of logical values and operations. Subtraction (b / d) representation. Fixed point combinator, expression of one of the functions: factorial, partial quotient, modulo remainder, or any other where recursion is required.

8. Measuring the complexity of the algorithm and the complexity of the problem. Classes P and NP. Reducibility of problems according to Karp. NP-completeness. Cook – Levin theorem: formulation and idea of the proof. Reducibility of the general satisfiability problem to the 3-CNF satisfiability problem and the reducibility of the 3-CNF satisfiability problem to one of the problems: clique, vertex cover, 3-coloring, Hamiltonian path, knapsack problem, integer linear programming - or another similar problem.

9. Basic rules of combinatorics: addition rule, multiplication rule. Dirichlet's principle. Inclusion-exclusion formula: proof, application to derive a formula for the number of disturbances. Basic combinatorial configurations: placements, permutations and combinations. Formulas for numbers of placements, permutations and combinations. Stirling's formula (b / d).

10. Binomial Newton formula, polynomial formula. Properties of binomial coefficients: symmetry, unimodality, recurrent formula of Pascal's triangle. An alternating sum of binomial coefficients. Estimates for binomial coefficients as $n \to \infty$: asymptotic behavior of n / k in the case k = const· n and in the case $k = o(\sqrt{n})$.

11. Formal power series: definition, operations on series (sum, difference, product, quotient, derivative). The Cauchy – Hadamard theorem on the radius of convergence (with proof). Generating functions: definition, examples of generating functions for a sequence of binomial coefficients and for Fibonacci numbers. An example of using generating functions to prove combinatorial identities.

12. Linear recurrence relations with constant coefficients (l.s.s.s.c.). Example: Fibonacci numbers. General view of the solution in an arbitrary case (b / d). Proof of the theorem on the general form of the solution of the l.s.s.s.s. second order (including for multiple roots of the characteristic polynomial). Application of the theorem to find a formula for the Fibonacci numbers.

13. Definition of a simple graph, digraph, multigraph, pseudograph, hypergraph. Routes in graphs, degrees of vertices. Graph isomorphism, graph homeomorphism. Graph planarity: definition of a planar graph, Euler's formula, upper bound for the number of edges in a planar graph. Pontryagin – Kuratovsky criterion (proof of necessity; sufficiency without proof). Euler and Hamiltonian cycles in graphs: a criterion for Euler, a sufficient condition for Hamiltonian (Dirac's theorem). Erdos-Khvatal sign (b / d).

14. Chromatic number, independence number, click number. Lower bound for the chromatic number in terms of the independence number and in terms of the click number; comparison of the order of these estimates for a random graph G (n, 1/2) in the Erdëos – Renyi model (a.s.s. \ alpha (G) <= 2log_2 (n)). Erdös' theorem on the existence of graphs with arbitrarily large girth and chromatic number.

15. Systems of common representatives (s.d.p.): definition, examples of problems that reduce to constructing an o.d.p .. Trivial upper and lower bounds for the size of a minimal o.p. , a theorem on the upper bound for the size of a "greedy s.r.p." The theorem on the unimprovability of this estimate in the general case (b / d).

16. Ramsey numbers: definition, and exact values of R (s, t) at $s \le 3$, $t \le 4$. Erdos – Szekeres upper bound, its corollary for the Ramsey diagonal numbers; lower bound for diagonal numbers using a simple probabilistic method.

17. Hypergraphs. T - intersection hypergraphs. Erdos – Ko – Rado theorem (on the maximum number of edges in a 1-intersection hypergraph). Fundamentals of the linear-algebraic method: the Frankl – Wilson theorem (an upper bound on m (n, r, s) in terms of the number of combinations for the case rs = p, p is prime, r <2p), a constructive lower bound on the Ramsey numbers (formulation, definition of a graph, doc -in the lemma on the independence number).

18. Probability space, Kolmogorov axioms, properties of a probability measure (including the theorem on the continuity of a probability measure with a proof). Conditional probabilities. Formula of total probability. Bayes' formula. Independence.

19. Random variables and vectors. Characteristics of a random variable and a vector: probability distribution, distribution function and its properties, σ-algebra generated by a random variable. Examples of specific distributions.

20. The mathematical expectation of a random variable: definition for simple, non-negative and arbitrary random variables. Basic properties of mathematical expectation (proofs only for prime values). Dispersion and covariance, their properties.
21. Convergence of random variables: in probability, in distribution, almost certainly, on average. Relationship between convergence (b / d). Slutsky's lemma (b / d). Convergence Inheritance Theorem. Delta method.
22. Markov inequality, Chebyshev inequality. The law of large numbers in the form of Chebyshev. Enhanced laws of large numbers (b / d).
23. Characteristic functions of random variables and vectors and their properties. Continuity theorem (b / d).
24. Central limit theorem for independent identically distributed random variables.
25. Sample, sample space. Point estimates of parameters and their main properties: unbiasedness, consistency, asymptotic normality. Sample mean, median, variance. Comparison of estimates, loss function and risk function. Approaches to comparing estimates: uniform, Bayesian, asymptotic.
26. Methods for constructing estimates: the method of moments and the method of maximum likelihood. Consistency of the evaluation of the method of moments. Theorem on the properties of maximum likelihood estimates (b / d).
27. Confidence intervals. Central statistics method. Method for constructing asymptotic confidence intervals.
28. Statistical hypotheses, errors of the first and second kind, the level of significance of the criterion. Criteria comparison principles are evenly the most powerful criteria. Neumann – Pearson lemma. Building with its help the most powerful criteria.
29. Dynamic array. Depreciation analysis. An estimate of the time it takes to add an element to a dynamic array (with a doubling of the filled buffer).
30. Linked lists. Stack, queue, decks and their implementations.
31. Quick sort (QuickSort). Search for ordinal statistics using the “Divide and Conquer” method (QuickSelect).
32. Merge sort (MeggeSort). Bitwise sorting.
33. Binary heap and heap sort (NotSort). Merge k sorted arrays using the heap.
34. Hash table, polynomial hash function.
35. Dynamic programming: general idea, linear dynamics, matrix dynamics, segment dynamics.
36. RMQ. Sparse table. Segment tree.
37. LCA: Reducing to RMQ and Binary Ascent Method.
38. Binary search tree. Depth and breadth crawls. Key search, naive key insertion and deletion. AVL tree. Red-black tree.
39. Cartesian tree. An implicit key Cartesian tree.
40. Minimum Spanning Tree: Prim and Kruskal's Algorithms.
41. Maximum network streams. Methods: Ford-Fulkerson; Edmonds-Karp (b / d).
42. Graph traversal in depth, width.
43. Finding the shortest paths in a graph: algorithms of Dijkstra, Ford-Bellman, Floyd-Warshall.
44. Search for strongly connected components in a graph.
45. Bridges and articulation points in the graph.
46. Finding a substring in a string: prefix function, Knuth-Morris-Pratt algorithm.
47. Standard containers: vector, deque, queue, priority_queue, set, map, iterators, comparators.
48. Bor. Aho-Korasik algorithm.
49. Suffix tree.
50. Calculation of the convex hull of a set of points in 2D and 3D.
51. Delaunay triangulation. Voronoi diagram.
52. Long arithmetic. Addition, subtraction, sign storage, multiplication, division. Karatsuba's algorithm.
53. Statement of the problem of supervised learning, statement of the problem of unsupervised learning. Statement of the classification problem and the classification quality metric. Naive Bayesian classifier.

54. Regression problem statement and regression quality metric. Linear regression. Gauss-Markov theorem (formulation). The problem of multicollinear features. L1 and L2 regularization, their influence on feature weights.

55. The problem of unbalanced classes. Working with categorical features and missing values. Mean encoding. Examples of simple algorithms that solve standard problems: kNN, a naive Bayesian classifier.

56. Logistic regression. Margin concepts. Equivalence of solutions obtained by the method of maximum likelihood and minimization of the logistic loss function. Logistic loss function, cross-entropy.

57. Solution of the classification problem by the support vector method. Margin concepts. Hinge-loss. Optimization task in SVM (declarative).

58. The procedure for constructing a decision tree, criteria for information content: entropy, Gini. Boosting (the principle of building an ensemble).

59. Bias-Variance decomposition (declarative). Bootstrap routine, bagging algorithm. Random Subspace Method (RSM), Random Forest.

60. Gradient boosting. The principle of construction. What algorithms can be used as basic algorithms.

61. Dimension reduction problem: PCA and t-SNE algorithms. Communication between PCA and SVD. Can the PCA provide more than one solution?

62. The problem of overfitting. Example, causes of occurrence (ideologically). Cross validation. Motivation for using train, val and test samples. The concept of model parameters and hyperparameters. Selection procedure for hyperparameters. The concept of regularization (in the general case). Regularization methods for various models (linear models, trees, ensembles, neural networks).

63. Gradient descent method. Backpropagation method. Activation functions (Sigmoid, tanh, ReLU), their properties and problems. Loss functions in the problem of multiclass classification and regression. Neural network regularization methods: Dropout, Batch normalization, data augmentation.

64. Neural network regularization methods: Dropout, Batch normalization, data augmentation. Weaknesses of Stochastic Gradient Descent Refinement methods: Momentum, Nesterov momentum, RMSprop, Adam. Disadvantages of these approaches.

65. Recurrent neural networks (RNN) for ordered data. Basic principles of work. The fading gradient problem and its possible solutions. Recurrent blocks: naive (Vanilla RNN), LSTM, GRU, motivation to use them. Activation functions in recurrent blocks.

66. Methods for working with images. Why are linear layers not widely used in computer vision? Convolutional layers in a neural network. Motivation for their use in the tasks of analyzing images and signals. One-dimensional and two-dimensional convolutions (Conv1d and Conv2d). Max & average pooling.

67. Nondeterministic finite state machines. Various definitions: single-letter transitions. Deterministic finite automata. Their equivalence.

68. Regular expressions. Kleene's theorem on the equivalence of regular expressions and finite automata.

69. Minimization of finite state machines. Minimization Algorithm. An algorithm for checking the equivalence of regular expressions.

70. Generative grammars. Chomsky hierarchy. Right-linear, context-free, context-sensitive grammars (definitions). Equivalence of right-linear grammars and finite state machines.

71. Context-free grammars. Chomsky normal form for context-free grammars.

72. Machines with push-in memory. Definition options. Equivalence of FFSMs and Context-Free Grammars: Constructing an Automaton from a Grammar.

73. Bullet-proof machines. Definition options. Equivalence of FFSMs and Context-Free Grammars: Constructing a Grammar from an Automaton.

74. Overgrowth lemmas for automaton and context-free languages. Examples of languages not included in these classes.

75. Parsing algorithms for context-free grammars. Coca-Younger-Kasami algorithm: description, complexity, correctness.

76. Parsing algorithms for context-free grammars. Earley's algorithm: description, complexity, correctness.

77. Operating systems and their components. Kernel of operating systems. System calls and their differences from ordinary library functions. Ways to implement system calls (interrupts, sysenter, syscall).

78. Integer arithmetic in computer representation. Signed and unsigned values, ways to represent negative values. Integer overflow and its control. Long integer arithmetic.
79. Real arithmetic. Fixed and floating point representations. IEEE754. Special real values defined by the IEEE754 standard and operations on them.
80. Processes and threads. The similarities and differences between them. Implementation of multitasking and task scheduling algorithms in operating systems.
81. Multithreaded synchronization problem. Atomic variables and blocking objects. Non-blocking data structures and their implementation.
82. Message Passing Interface (MPI) Existing implementations, tasks of MPI as a programming environment. Life cycle of an MPI program. Creation and termination of MPI processes. Organization of the I / O stream, parameters specified to the MPI program.
83. The concept of acceleration and scalability of parallel programs. Amdahl's Law. Evaluation of the effectiveness of parallel programs. Layered parallel form of the program.
84. Distributed file systems. Roles of system elements, ensuring fault tolerance. Read and write algorithms in distributed file systems. Data replication.
85. The computation model MapReduce. Key-value pairs in MapReduce implementations. The main stages of calculations and additional elements of the model.
86. Data join (Join operation) in a MapReduce model. Computing and data optimization model.
87. Iterative calculations on large amounts of data. The lazy evaluation model and data storage structure in the Spark implementation (RDD). Calculation results caching and iterative calculations.
88. Distributed message managers. Replication and implementation of fault tolerance. Semantics of message delivery.
89. Transaction isolation levels. The principles of atomicity, consistency, isolation and resilience (ACID rules). Committing and canceling transactions.
90. Fisher-Lynch-Paterson theorem (FLP-theorem), CAP-theorem and their applications. Distributed configuration storage systems.
91. The choice of a leader process or a leader machine in distributed systems. Consensus Algorithm.

4. Procedure of taking a state examination

The state exam is admitted to a student who has mastered the disciplines that are covered by the exam program, and does not have academic debts for them.

The state exam consists of two parts: 1) in the disciplines from the field of mathematics (sections 1-3 of the exam program); on disciplines from the field of algorithms and computer technology (sections 4-7 of the exam program). Each part of the exam has a separate day.

The exam format is oral.

1. The order of delivery of a part in discrete mathematics

The exam ticket consists of two questions, the choice of which is carried out by students from sections [1-3], on one question from two different sections. A survey of students on the choice of optional sections is carried out at least one week before the date of the state exam
Examination tickets are cut into several parts, grouped according to sections of the exam, the student pulls different parts of the ticket from the corresponding piles. The student is given 1 astronomical hour to prepare for the oral answer. During preparation for the answer, it is allowed to use the "official cheat sheet", course materials.
The oral part of the exam includes the student's answer to the questions of the examination card. After completing the oral response, GEC members can ask additional and clarifying questions, including those requiring additional preparation time.

2. The order of delivery of the part on algorithms and computer technologies

Examination tickets consist of three questions: one from the obligatory section [4], and two questions from sections [5,6 or 7] at the discretion of the student. A survey of students on the choice of optional sections is carried out at least one week before the date of the state exam.

Examination tickets are cut into several parts, grouped according to sections of the exam, the student pulls different parts of the ticket from the corresponding piles. The student is given 1.5 astronomical hours to prepare for the oral answer. During preparation for the answer, it is allowed to use previously prepared handwritten notes without limitation in volume.

The oral part of the exam includes the student's answer to the questions of the examination card. After completing the oral response, GEC members can ask additional and clarifying questions, including those that require additional preparation time.

3. Peculiarities of passing the exam in conditions of restrictions associated with COVID-19

In the event of restrictive measures prohibiting the conduct of the exam in public, the exam is conducted using remote means that provide the possibility of video communication.

Before the start of the exam and receipt of the ticket, the student must provide (show on camera) an identity document: a record book, student card, or passport.

During the preparation for the answer, it is allowed to use any sources of information, including electronic ones, but the preparation time is reduced to 30 minutes.

After completing the oral answer, the members of the SEC are obliged to ask one additional question from each section of the exam program, for which the student was preparing. Additional questions do not involve a long preparation time and require a short answer.

5. Description of the facilities required for the state examination

Auditorium for the certification test, equipped with places for students and the examination committee.
In the event of restrictive measures prohibiting the conduct of the exam in a public form - workable remote workplaces for all students and examiners.

6. List of recommended reading

Main literature

1. Программирование: теоремы и задачи [Текст], [учеб. пособие] /А. Шень. -М., МЦНМО, 2011

2. Математическая теория формальных языков / А. Е. Пентус, М. Р. Пентус. — Москва, ИНТУИТ, 2016.— URL: https://ibooks.ru/bookshelf/362911/reading (дата обращения: 26.11.2020). - Полный текст (Режим доступа : из сети МФТИ / Удаленный доступ)

Additional literature

1. Архитектура компьютера, [учеб. пособие для вузов] / Э. Таненбаум, Т. Остин. — Санкт-Петербург, Питер, 2020.— URL: https://ibooks.ru/bookshelf/361850/reading (дата обращения: 26.11.2020). - Полный текст (Режим доступа : из сети МФТИ / Удаленный доступ)

7. Guidelines for students preparing for the state examination

When preparing for the oral part of the state exam, students are encouraged to recall the topics of mathematical disciplines included in the program of the oral part of the state exam, using lecture notes and recommended literature, if necessary. After repeating each topic, the student is advised to independently write the formulations and proofs of the theorems contained in the program of the oral part of the state exam, without using literature and aids.

If there are questions that the student cannot independently solve with the help of the recommended literature, it is recommended to ask these questions at the consultation conducted by the teacher of the department in the relevant discipline.

8. Methodology and assessment criteria for the state examination

For each of the two parts of the exam, a separate mark is given on a 10-point scale. Grades below 3 are considered unsatisfactory and blocking: if you receive an unsatisfactory mark for any part of the exam, the exam is considered passed for the grade "unsatisfactory", and the student is not allowed to pass the other part of the exam.

Answers to the oral part for each of the sections of the program are assessed separately on a 10-point scale. The mark for the part of the exam that is held on the same day is determined as the arithmetic mean of the marks for the different sections. The result of the assessment for the part is rounded to the nearest whole value according to the standard arithmetic rules and is announced to the student on the day of the part of the exam.

The final grade is set on a 10-point scale and is defined as the average of two grades obtained for different parts of the exam. The rounding method is standard arithmetic.

excellent (10) - correct, clear and confident answer to both questions of the ticket and additional questions;

excellent (9) - correct answers were given to both questions of the ticket and additional questions with minor inaccuracies;

excellent (8) - answers to both questions of the ticket and additional questions are given after minor corrections and leading questions from the examiners;

good (7) - answers to both questions of the ticket are given, but there is no correct answer to one of the additional questions;

good (6) - there are flaws in the answer to one of the questions on the ticket and there is no correct answer to one of the additional questions;

good (5) - there are gaps in the answers to both questions of the ticket and there is no correct answer to one of the additional questions;

satisfactory (4) - there are gaps in the answers to both questions of the ticket or there is no answer to any of the additional questions;

satisfactory (3) - there is no answer to one of the questions on the ticket, but there are answers to additional questions (possibly with deficiencies);

unsatisfactory (2) - there is no answer to one of the ticket questions and to additional questions;

unsatisfactory (1) - there is no answer to any of the ticket questions.

9. Peculiarities of state final examinations for persons with disabilities and persons with special needs

For students with disabilities, the final state assessment takes into account the particularities of their psycho-physical development, their individual capacities and their state of health (hereinafter referred to as the individual characteristics).

9.1. The following general requirements shall be ensured in the conduct of the FSA:

– conducting state final examinations for persons with disabilities in the same room as students without disabilities, if this does not create difficulties for the students when taking the final examinations;

– presence of assistant(s) in the classroom to provide students with disabilities with the necessary technical assistance, taking into account their individual characteristics (to take the workplace, move around, read and complete an assignment, communicate with members of the SEC);

– the use of technical aids for students with disabilities in taking the FSA, taking into account their individual characteristics;

– ensuring that students with disabilities have unhindered access to and use of classrooms, toilets and other facilities.

9.2. At the written request of a student with a disability, the duration of the state certification test may be extended beyond the established duration of the test:

– the duration of the written state examination - not exceeding 90 minutes;

– the duration of the preparation of the student's answer to a state examination held orally - not more than 20 minutes.

9.3. A student with a disability shall submit a written application no later than 3 months prior to the commencement of the State Attestation Examination regarding the need to create special conditions for him/her when conducting state attestation tests, indicating the specifics of his/her psychophysical development, individual capabilities and state of health. The application shall be accompanied by documents confirming the learner's individual characteristics (in the absence of these documents from the Institute Directorate).

In the application, the student shall indicate the need (lack of need) for the assistant's presence at the state attestation test, the need (lack of need) for increasing the duration of the state attestation test in relation to the established duration.

10. Examples of control tasks, tickets

Examples of tasks are given in the supplement.

10. Examples of test assignments, tickets

10.1. Examples of tickets for part of the exam by sections [1-3]

Ticket number 1

1. The concept of a set and a subset. Operations on sets, identities. Mappings and matching. Comparison of sets by cardinality. Cantor – Bernstein theorem. Countable sets and their properties. Cantor's theorem.

2. Chromatic number, independence number, click number. Lower bound for the chromatic number in terms of the independence number and in terms of the click number; comparison of the order of these estimates for the random graph G (n, 12) in the Erdëos – Renyi model (a.s.s. \ alpha (G) <= 2log_2 (n)). Erdös' theorem on the existence of graphs with arbitrarily large girth and chromatic number.

Ticket number 2

1. Ramsey numbers: definition, and exact values of R (s, t) at s3, t4. Erdos – Szekeres upper bound, its corollary for the Ramsey diagonal numbers; lower bound for diagonal numbers using a simple probabilistic method.

2. Walkability of random variables: by probability, by distribution, almost surely, on average. Relationship between convergence (b / d). Slutsky's lemma (b / d). Convergence Inheritance Theorem. Delta method.

Ticket number 3

1. Confidence intervals. Central statistics method. A method for constructing asymptotic confidence intervals.

2. Formal arithmetic. Examples of conclusions in Peano's axiomatics. Modeling Turing machines in formal arithmetic (b / d). Gödel's Incompleteness Theorem: Statement and Idea of Proof.

10.2. Examples of tickets for a part of the exam by sections [4-7]

Ticket number 1

Graph traversal in depth, width.

The problem of overfitting. Example, causes of occurrence (ideologically). Cross validation. Motivation for using train, val and test samples. The concept of parameters and hyperparameters of models. Selection procedure for hyperparameters. The concept of regularization (in the general case). Regularization methods for various models (linear models, trees, ensembles, neural networks).

Fisher-Lynch-Paterson theorem (FLP-theorem), CAP-theorem and their applications. Distributed configuration storage systems.

Ticket number 2

Minimal Spanning Tree: Prim and Kruskal's Algorithms.

Solution of the classification problem by the support vector method. Margin concepts. Hinge-loss. Optimization task in SVM (declarative).

Bullet-proof machines. Definition options. Equivalence of FFSMs and Context-Free Grammars: Constructing a Grammar from an Automaton.

Ticket number 3

1.LCA: Reducing to RMQ and Binary Ascent Method.

2. Dimension reduction problem: PCA and t-SNE algorithms. Communication between PCA and SVD. Can the PCA provide more than one solution?

3. Overgrowth lemmas for automaton and context-free languages. Examples of languages not included in these classes.