THE PROGRAM OF THE ENTRANCE TEST IN COMPUTER SCIENCE FOR THE STUDENTS ENTERING MASTER PROGRAM

Entrance test consist of writing part (it takes 2 hour), the results of which are rated.

1. Algorithms

1.1. The concept of algorithm.
1.2. The concept of spatial complexity of the algorithm.
1.3. The concept of time complexity of the algorithm.
1.4. Turing’s machine.
1.5. Post’s Machine.
1.6. Normal Markov’s algorithms.
1.7. Halting problem.
1.8. Sorting algorithm.
1.9. Greedy algorithm.
1.10. Binary search.
1.11. Dynamic programming (2, 3-dimensional)
1.12. Dynamic programming on sub-segments.
1.13. Dynamic programming by profile.
1.14. Discrete and continuous "knapsack" problem.
1.15. The least Common ancestor (LCA) problem.
1.16. Complexity classes of algorithms (P, NP).
1.17. The prefix-function.
1.18. Z-function. Knuth-Morris-Pratt’s algorithm.
1.19. Aho-Corasick’s algorithm.
1.20. The extended Euclidean algorithm.
1.21. Eratosthenes’s sieve.
1.22. Lossless data compression algorithms.

2. Data structure

2.1. Stack
2.2. Queue
2.3. Priority queue
2.4. Decks
2.5. Single-linked lists.
2.6. Doubly linked lists.
2.7. Heap.
2.8. Binary tree.
2.9. Cartesian tree.
2.10. Hash tables.
2.11. Binary container (Range Minimum Query, RMQ), description and examples of applications.
2.12. The suffix array.
2.13. Suffix automaton.
2.14. The tree of segments.
2.15. Fenwick Tree.

2.16. Red-black wood.

3. C++programming language

3.1. Preprocessor.
3.2. Branches
3.3. Cycles
3.4. Functions
3.5. Arrays
3.6. Pointer arithmetic
3.7. Recursion.
3.8. Structures.
3.9. Association (union).
3.10. The standard library of C.
3.11. STL library.
3.12. Boost.

4. Java programming language

4.1. Java virtual machine.
4.2. Memory management.
4.3. Passing primitive types to functions.
4.4. Passing reference types to functions.
4.5. The problem of changing reference within a subroutine.
4.6. Static initializer.
4.7. Deletion of unused objects and finalize method.
4.8. The problem of destructors for complex objects.
4.9. Garbage collection

5. Electronic Computer Machine Architecture

5.1. Computer architecture (Harvard, von Neumann)
5.2. The set of commands of the processor (CISC, RISC, VLIW)
5.3. Cache and acceleration of work with its use.
5.4. Convention about calling.
5.5. Representation of integers. Additional code.
5.6. Representation of floating point numbers.
5.7. Assembly language.
5.8. Reverse engineering.
5.9. Debugging and instrumentation tools (valgrind, AddressSanitizer, strace, gdb)
5.10. Static and dynamic libraries.

6. Operating system

6.1. Classification of operating systems.
6.2. Real-time operating systems.

6.3. The concept of process, types of processes.
6.4. Files. File system structure.
6.5. Memory management: single allocation, paged, segmented, segment-page, swapping.
6.6. Process interaction, IPC: pipes, signals, message queues, sockets, semaphores, shared memory.
6.7. Users and groups.
6.8. Mandatory access control.
6.9. Types of virtualization.

7. Object-oriented programming.

7.1. The concept of object and class of objects.
7.2. Encapsulation
7.3. Inheritance
7.4. Polymorphism
7.5. Design
7.6. Destructor
7.7. Templates
7.8. Exceptions and their handling.

8. Computational mathematics

8.1. Calculation error.
8.2. Changing the error in arithmetic operations.
8.3. Methods of error reduction in calculations.
8.4. Newton method.
8.5. The concept of calculated grid.
8.6. Ternary search.
8.7. Gauss method.
8.8. Linear programming.
8.9. Rapid exponentiation.
8.10. Long and modular arithmetic.
8.11. Fast multiplication of polynomials.
8.12. Sparse representation of matrices. Fast multiplication of matrices.
8.13. The method of trapezoids.
8.14. The concept of computational experiment.

9. Parallel programming

9.1. Superscalar architecture
9.2. SISD, SIMD, MIMD architectures.
9.3. Message passing (mpi)
9.4. Working with shared memory (openMP)
9.5. Parallelization on graphics accelerators (CUDA, OpenMP)
9.6. The concept of mutual blocking (deadlock).

10. Networks

10.1. The ISO/OSI model
10.2. IPv4 Protocol. The concept of IP address, subnet mask. IPv6.
10.3. System calls for network support in OS (socket, bind, listen, access, connect,read, write, send, recv...).
10.4. TCP and UDP protocols.
10.5. Serialization / deserialization.
10.6. Basics of the HTML language. Basic tags.
10.7. Domain name system.
10.8. The concept of network latency, RTT.
10.9. Remote procedure call.

11. Computer graphics

11.1. Representation of color in the computer.
11.2. Graphic format.
11.3. Vector and raster formats.
11.4. Projections
11.5. The method of marching cubes.
11.6. Fourier’s fast transformation.
11.7. Data compression with loss of quality.

12. Artificial intelligence

12.1. Machine learning. The concept of training and control samples.
12.2. Neuronets.
12.3. 3 laws of robotics.

13. Methods of data analysis and recognition

13.1. Recognition problem.
13.2. Classification problem.
13.3. Errors of the first and second kind.

14. Graph theory.

14.1. Tops
14.2. Ribs
14.3. Adjacency matrix.
14.4. Incidence matrix
14.5. Tree.
14.6. Cayley formula (number of spanning trees in a complete graph).
14.7. Bypass in depth.
14.8. Bypass in width.
14.9. The flow in the graph.
14.10. Routes, chains, cycles

14.11. Euler's way.
14.12. Hamilton's way.
14.13. Floyd's Algorithm
14.14. Dijkstra's Algorithm
14.15. Kruskal's Algorithm
14.16. Dinitz’s Algorithm.
14.17. Bipartite graphs. Pairs.
14.18. The planarity of the graph.

15. Databases

15.1. Database classification by data model.
15.2. Relational theory.
15.3. Attributes, tuples, domains, relationships.
15.4. Primary and foreign keys.
15.5. Normal forms.
15.6. Relational operations.
15.7. Aggregates, grouping, and analytical functions.
15.8. Physical content of database. Data page.
15.9. The basics of the SQL language. The queries SELECT, INSERT, UPDATE.

16. SOFTWARE development and project management

16.1. Automation of program Assembly. The Make Utility.
16.2. The concept of technical specifications.
16.3. The principles of testing. Classification of defects. Test-driven-development.
16.4. Version control systems: cvs, svn, mercurial, git.
16.5. Quality management system.
16.6. Methods of structural design. Types of methods: top-down, top-down, iterative. Modularity. Principles of division of the system into modules. Quality metrics of modular structure. The method of gradual refinement, structural diagrams (STD). Data flow diagrams (DFD). Method Jackson structural programming (JSP).

17. Information security.

17.1. Concepts of vulnerability and exploit. "Black "and" White" hackers.
17.2. Legislation in the field of information security.
17.3. Symmetric cryptography.
17.4. Asymmetric cryptography.
17.5. The criteria of simplicity. Connection of simple numbers and cryptography.
17.6. Random number generation. Random and pseudorandom numbers. Reproducibility.
17.7. Buffer overflow.
17.8. Stack overflow.
17.9. Executing arbitrary code on the server side.
17.10. SQL-injections.
17.11. Attacks on wireless networks.

18. Algebra of logic

18.1. Logical variables.
18.2. Basic operation of algebra of logic (negation, conjunction, disjunction, exclusive or.)
18.3. Truth table.
18.4. Completeness of the system of functions.

19. Theory of formal languages.

19.1. The concept of language.
19.2. Formal grammar.
19.3. Context-free grammar.
19.4. Context-sensitive grammar.
19.5. State machine.
19.6. Vending machines.

20. Computational geometry

20.1. The concept of point and vector. Relevant data structures.
20.2. Scalar vector product.
20.3. Vector product.
20.4. The oriented area of the triangle. The area of an arbitrary primary polygon.
20.5. "Clockwise" predicate. Testing segments for intersection without computing the point of intersection.
20.6. The distance from a point to a straight line, from a point to a section.
20.7. Finding point of intersection of two lines. The normal equation of a straight line.
20.8. The intersection of a circle and a straight line. The intersection of two circles.
20.9. Convex hull (with complexity O (N log N)).
20.10. Method of scanning straight line.

21. Classification of programming languages.

21.1. Procedure languages.
21.2. Logic languages.
21.3. Functional languages.
21.4. Markup languages (XML, TeX).

Literature for self-study

1. Bruce Eckel. Thinking In C++;
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms, 3rd Edition (The MIT Press);
3. Brian Kernighan and Dennis Ritchie. The C Programming Language;
4. Scott Meyers. Effective STL;
5. Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman. Data Structures and Algorithms;
6. Jon Louis Bentley. Programming Pearls;
7. N. Wirth. Algorithms and Data Structures;
8. Nicholas A. Solter, Scott J. Kleper. Professional C++;

9. Herbert Schildt. C++: The Complete Reference;
10. Robert Sedgewick. Algorithms in C++;
11. Martin Fowler. UML. Distilled. A Brief Guide to the Standard Object Modeling Language. Third Edition;
12. Craig Larman. Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design and the Unified Process;
13. Erich Gamma, Ralph Johnson, Richard Helm, John Vlissides. Design Patterns. Elements of Reusable Object-Oriented Software;
14. R. Pressman. Software Engineering: A Practitioner's Approach, 6th Ed.;
15. L. Bass, P. Clements, R. Kazman. Software architecture in practice;
16. Scott W. Ambler, Ron Jeffries. Agile Modeling: Effective Practices for Extreme Programming and the Unified Process;
17. Bertrand Meyer. Object-Oriented Software Construction;
18. B.Liskov, J.Guttag. Program Development in Java: Abstraction, Specification and Object-Oriented Design;
19. Bruce Eckel. Thinking in Java, 4th Edition;
20. Andrew S. Tanenbaum, Herbert Bos. Modern Operating Systems4
21. W. Richard Stevens. UNIX Network Programming;
22. W. Richard Stevens, Stephen A. Rago. Advanced Programming in the UNIX Environment;
23. C.J.Date. An Introduction to Database Systems.