Open Informatics Master State Exam Topics Specification

A more detailed specification of the accredited state exam topics"

2021-05-28

- The accredited state exam topics (in English)
- The accredited state exam topics (in Czech)

Contents

Common part

1. **Polynomial algorithms for standard graph problems. Combinatorial and number-theoretical algorithms, isomorphism, prime numbers. Search trees and their use. Text search based on finite automata.** BE4M33PAL (*Course web pages*)

- Notation of asymptotic complexity of algorithms. Basic notation of graph problems - degree, path, circuit, cycle. Graph representations by adjacency, distance, Laplacian and incidence matrices. Adjacency list representation.
- Algorithms for minimum spanning tree (Prim-Jarník, Kruskal, Borůvka), strongly connected components (Kosaraju-Sharir, Tarjan), Euler trail. Union-find problem. Graph isomorphism, tree isomorphism.
- Generation and enumeration of combinatorial objects - subsets, k-element subsets, permutations. Gray codes. Prime numbers, sieve of Eratosthenes. Pseudorandom numbers properties. Linear congruential generator.
- Search trees - data structures, operations, and their complexities. Binary tree, AVL tree, red-black tree (RB-tree), B-tree and B+ tree, splay tree, k-d tree. Nearest neighbor searching in k-d trees. Skip list.
- Finite automata, regular expressions, operations over regular languages. Bit representation of nondeterministic finite automata. Text search algorithms - exact pattern matching, approximate pattern matching (Hamming and Levenshtein distance), dictionary automata.

2. **Problem/language complexity classes with respect to the time complexity of their solution and memory complexity including undecidable problems/languages.** BE4M01TAL (*Course web pages*)

- Asymptotic growth of functions, time and space complexity of algorithms. Correctness of algorithms - variant and invariant.
- Deterministic Turing machines, multitape Turing machines, and Nondeterministic Turing machines.
- Decision problems and languages. Complexity classes P, NP, co-NP. Reduction and polynomial reduction, class NPC. Cook theorem. Heuristics and approximate algorithms for solving NP complete problems.
- Classes based on space complexity: PSPACE and NPSPACE. Savitch Theorem.
- Randomized algorithms. Randomized Turing machines. Classes based on randomization: RP, ZPP, co-RP.
- Decidability and undecidability. Recursive and recursively enumerable languages. Diagonal language. Universal language and Universal Turing machine.

3. **Combinatorial optimization problems - formulation, complexity analysis, algorithms and example applications.** BE4M35KO (*Course web pages*)

- Integer Linear Programming. Shortest paths problem and traveling salesman problem ILP formulations. Branch and Bound algorithm. Problem formulations using ILP. Special ILP problems solvable in polynomial time.
- Shortest paths problem. Dijkstra, Bellman-Ford, and Floyd–Warshall algorithms. Shortest paths in directed acyclic graphs. Problem formulations using shortest paths.
- Network flows. Maximum flow and minimum cut problems. Ford-Fulkerson algorithm. Feasible flow with balances. Minimum cost flow and cycle-canceling algorithm. Problem formulations using network flows. Maximum cardinality matching.
- Knapsack problem. Approximation algorithm, dynamic programming approach, approximation scheme.

- Traveling salesman problem. Double-tree algorithm and Christofides algorithm for the metric problem. Local search k-OPT.
- Scheduling - problem description and notation. One resource - Bratley algorithm, Horn algorithm. Parallel identical resources - list scheduling, dynamic programming. Project scheduling with temporal constraints - relative order and time-indexed ILP formulations.
- Constraint Satisfaction Problem. AC3 algorithm.

Artificial Intelligence

1. **Learnability models: PAC and online. Learnability of conjunctions and disjunctions. Bayesian networks. Reinforcement learning.** BE4M36SMU (*Course web pages*)

- PAC and online learnability models: definition, efficient learnability. Comparison of the two models: differences in assumptions (such as i.i.d. observations), mutual relations (does one imply the other?). Vapnik-Chervonenkis dimension. Necessary and sufficient conditions for learnability.
- If one or the other concept class is learnable in one or the other learnability models, show an algorithm that learns it in that model. Are they also learnable efficiently? Learning other hypothesis classes by reduction to learning conjunctions or disjunctions.
- Bayesian networks: the notion of conditional independence and how BN defines it for a set of random variables. Computing conditional probabilities and the most probable state from a BN. Estimating BN parameters through maximum likelihood.
- Reinforcement learning: state utility, optimal policy, value iteration, direct utility estimation, adaptive dynamic programming, temporal difference learning, exploration vs. exploitation, Q-learning, and SARSA. Policy search.

2. **Resolution in the first order logic, automatic proving. Principles of automatic proving in Boolean domains and in predicate logic. Searching for models in generic domains.** BE4M36LUP (*Course web pages*)

- Normal forms in propositional logic. The Boolean satisfiability problem (SAT) and basic algorithms: the resolution rule, unit propagation, and clause learning. The Satisfiability Modulo Theories problem and the lazy approach.
- Logic programming and Prolog: definite and Horn clauses, Herbrand interpretations, minimal model semantics, negation as failure, SLDNF resolution, cut.
- Normal forms in first-order logic (FOL). The resolution calculus in FOL: the purpose of unification in it, subsumptions, and the saturation procedure. The handling of equality in FOL: axiomatic approach vs. extending the resolution rule.
- Searching for models in generic domains: problem grounding and propositional encoding as the SAT problem.

3. **Minimizing empirical risk. Maximum likelihood estimation, EM algorithm. Deep networks and their training. Classical and deep neural networks and their learning.** BE4M33SSU (*Course web pages*)

- Empirical risk minimization: Risk and empirical risk of a predictor, generalization bounds and Hoeffding inequality, statistically consistent learning algorithms, Vapnik-Chervonenkis-dimension of a hypothesis class, SVMs and Kernel SVMs.
- Maximum likelihood estimators and the EM algorithm: maximum likelihood estimator, consistency of an estimator, EM algorithm as maximization of a lower bound of the likelihood, E-step and M-step as block-coordinate descent for the lower bound.
- Deep networks, supervised learning of networks: neurons, network architectures, convolutional networks, backpropagation and layer types, parameter initialisation, stochastic gradient descent.

4. **Domain independent planning. Features, heuristics and algorithms.** BE4M36PUI (*Course web pages*)

- STRIPS and SAS representation of planning tasks. Satisficing and optimal planning. Heuristic search. Properties of heuristic functions.
- Delete-relaxation heuristics. h_{max}, h_{add}, and h_{ff} heuristics. Abstraction heuristics. Projection, pattern databases. Merge & Shrink heuristic.

- Landmarks and landmark discovery. Landmark and LM-Cut heuristics.
- Linear Programming heuristics. State-Equation heuristic. Potential heuristic.
- Non-deterministic planning. Probabilistic planning. Markov Decision Process and Value Iteration. Monte-Carlo Tree Search and the UCT algorithm.

5. **Autonomous agents and multiagent systems. Noncooperative game theory.** BE4M36MAS (*Course web pages*)

- Normal-form games (NFGs) and the concept of mixed strategy. Nash equilibrium - its properties and computation. Removal of dominated strategies. Two-person zero-sum games and the algorithm based on linear programming. Alternatives to Nash equilibrium for NFGs: correlated equilibrium and Stackelberg equilibrium.
- Extensive-form games (EFGs), their representation, and properties (imperfect information and perfect recall). Types of strategies in EFGs. The concept of Nash and subgame perfect equilibrium. How to solve EFGs: the algorithms based on linear programming and on regret minimization.
- Coalitional Games and their representation. Basic classes and examples of coalitional games. The core of a game and its properties. Axioms of fair allocation for the Shapley value. The computation of the Shapley value. Simple voting games and their solution by the Shapley-Shubik index and by the Banzhaf index.
- Auction mechanisms and their properties. How to bid in single-item auctions. Strategic and revenue equivalence of auction mechanisms. Optimum single-item auctions. Combinatorial auctions and their representations.

6. **Decision making, planning and coordination of autonomous systems of one or more robots.** BE4M36UIR (*Course web pages*)

- Robotic paradigms and control architectures, embodiment, sensor, and actuators. Properties of hierarchical, reactive, and hybrid paradigms. Their advantages and drawbacks.
- Path and motion planning problem formulations. Notation of the configuration space and roadmap-based planning methods. Variants of the graph-based planning, existing speedup techniques, and planning approaches for environments that can dynamically change such as D* or D* Lite algorithms.
- Informative path planning methods and robotic exploration of unknown environments - problem decomposition and multi-robot task allocation algorithms - centralized and decentralized methods. Variants of frontier-based exploration and information theory-based methods.
- Multi-goal planning and robotic variants of the sequence-dependent traveling salesman problem. Existing problem formulations and extensions with curvature-constrained multi-goal trajectory. Formulations of the routing problems with profits in data collection planning scenarios. Decoupled, transformation, and sampling-based solution methods.
- Sampling-based motion planning methods - probabilistic completeness and asymptotic optimality.

Computer graphics

1. **Raster graphic. 3D objects and 3D scenes, transformations. Visibility, local illumination methods, shading and shadows. Radiometry, global illumination methods, texturing.**

   BE4M39APG (*Course web pages*)

- Raster graphics. Line drawing algorithms: DDA, Bresenham Algorithm. Filling algorithms: Polygon filling, Raster seed-based algorithms. Clipping lines and polygons.
- 3D object representations: boundary and volumetric representations, mesh representation (geometrical and topological data and their efficient encoding), representing a 3D scene using scene graph.
- Transformations: representing transformations (matrices, quaternions), composing transformations, applying transformations, associated computational costs.
- Visibility algorithms: z-buffer algorithm (principle, properties, issues), painter's algorithm, BSP tree ordering, ray casting and its comparison to z-buffer.
- Illumination methods: radiometry (radiometric quantities, their units and relations), reflectance equation, BRDF, BRDF models (Phong, Cook-Torrance), rendering equation, Whitted ray tracing, path tracing, radiosity.
- Shadow computation algorithms: ray cast shadows, the shadow map algorithm.
- Texturing: principles of texture mapping, uv maps, texture types (material textures, bump maps, normal maps, alpha masks), MIP mapping.

2. **Data structures for searching in multidimensional spaces.** BE4M39DPG (*Course web pages*)

- Regular and hierarchical data structures, time and space complexity. Cost model and general building and search algorithm for hierarchical data structures.
- Data structures for representations of points and shapes including nets, special data structures (2.5D, voxel-based representations, point-based data structures).
- Data structures and applications for the nearest neighbor, k-nearest neighbor and range search queries. Data structures and algorithms for high-dimensional search.
- Algorithms for building and traversal of the data structures for ray shooting and applications.
- Data structures for collision detection, the complexity and applications.

3. **Objects representation and their animation. Tools to support the production process.** BE4M39MMA (*Course web pages*)

- The notion of animation and animation loop. Representations of selected objects in 3D/2D vector space and algorithms for their animation: forward and inverse kinematics, solution of IK problem.
- Dynamics, particle systems, force interactions; facial animation; cloth; smoke, selected algorithms of fluid simulation; crowd.
- Production process, techniques and technologies: livecycle of computer animation project; multimedia archivation problem.
- Motion description techniques, motion capture principles, motion data processing methods; techniques for presentation of stereoscopic content, light control systems.

4. **Basic data structures of computational geometry, methods of their construction, and representation.** BE4M39VG (*Course web pages*)

- Convex set, convex hull of a set (definitions). Convex hull (2D) representation. Its computation from a set of points: Graham's Algorithm, Jarvis' Algorithm of 'gift wrapping', Quick hull, and the Divide and Conquer method,. Computation of convex hull for a simple polygon. Computation and representation of convex hull in 3D. Complexity, dependency on the distribution of points and on the number of points on the convex hull.

- Point location in a polygon and in a planar subdivision (the method of slabs, the tree of monotonous chains, trapezoidal map). Representation of planar subdivision (DCEL).
- Voronoi Diagram (VD), Delaunay triangulation (DT), and their relation. VD computation from a set of points via the Incremental, Divide and Conquer, and Fortune's algorithm. DT computation by the Incremental method, edge legalization via edge-flip operation, maximization of angular sequence.

5. **Scientific visualization methods. Information visualization methods.** BE4M39VIZ (*Course web pages*)
- Characteristics of spatial data and spatial fields. Grids used to represent spatial fields and their relation with data enrichment. Visualization of scalar fields with color mapping and contouring. Direct volume rendering.
- Visualization of vector fields with glyphs, stream objects, and line integral convolution (LIC).
- Characteristics of abstract data. Visualization of tabular (or n-dimensional) data. Visualization of relational data (hierarchies, directed acyclic graphs, and undirected graphs).
- Visualization of text. Visualization of a single document. Visualization of document corpus.
- Visualization of time-varying data. Time primitives and relations between them. Time domain, its type, granularity and structure.

6. **Spatial geometry, image projection and perspective camera model for 3D reconstruction, virtual reality, visual odometry and SLAM.** BE4M33GVG (*Course web pages*)
- Affine and projective plane and space. Representation of points, lines, planes, angles and distances and elementary operations with these entities.
- Mathematical model of a perspective camera: projection matrix. Calibration of camera intrinsic parameters from known objects, camera resection from six points, exterior orientation from three points, and the corresponding algorithms.
- Camera pair models: Homography. Epipolar geometry and epipolar constraint.
- 3D scene reconstruction from images. Homography in P2 and the 4-point algorithm. Fundamental matrix and the 7-point algorithm. Essential matrix and the 5-point algorithm.
- Computing camera motion and scene structure from image correspondences. Reprojection error and its Sampson approximation. Triangulation with Sampson correction. Bundle adjustment, gauge freedom and minimal representations.

Human-Computer Interaction

1. **Scientific visualization methods. Information visualization methods.** BE4M39VIZ (*Course web pages*)
   - Characteristics of spatial data and spatial fields. Grids used to represent spatial fields and their relation with data enrichment. Visualization of scalar fields with color mapping and contouring. Direct volume rendering.
   - Visualization of vector fields with glyphs, stream objects, and line integral convolution (LIC).
   - Characteristics of abstract data. Visualization of tabular (or n-dimensional) data. Visualization of relational data (hierarchies, directed acyclic graphs, and undirected graphs).
   - Visualization of text. Visualization of a single document. Visualization of document corpus.
   - Visualization of time-varying data. Time primitives and relations between them. Time domain, its type, granularity and structure.

2. **A formal description of user interfaces. Models of human behavior in relation to user interaction. Formal evaluation and prototyping.** BE4M39NUR (*Course web pages*)
   - Formal models of an interactive system (e.g., HMVC, Arch, PAC, Seeheim). Formal models of a user interface (CTT, STN).
   - Formal models of human behavior (HMP, GOMS, KLM). Laws describing human performance (Fitts's law, Hick-Hyman law). Basic cognitive processes related to human-computer interaction.
   - Formative evaluation of user interaction with an interactive system. Prototyping techniques (LoFi, HiFi, Wizard of Oz). User research and methods for description of user needs and behavior (scenarios, storyboards, HTA).

3. **User research and its role in HCI. Cognitive psychological concepts and their usage in HCI.** BE4M39PUR1 (*Course web pages*)
   - User research strategies (qualitative and quantitative; generative and evaluative), data collection methods (usability testing, interviews, observation, questionnaire, ethnography, etc.), methodological and research concepts (validity, reliability, research sample representativeness, etc.), sampling strategies.
   - Basics of human cognition applied in HCI (visual perception, memory, cognitive styles, attention, CLT, concept of flow, irrationality).
   - Psychological topics related to HCI beyond human cognition (persuasiveness, emotions in design, concept of beauty and its relation to usability).

4. **Statistical analysis, models and their assessment. Dimensionality reduction. Clustering.** BE4M36SAN (*Course web pages*)
   - Multiple linear regression. Describe the model and its assumptions. Treatment of qualitative independent variables, collinearity and outliers. Decide whether a model is useful. Overfitting, feature selection, model regularization.
   - Non-linear regression, polynomial regression, splines, local regression.
   - Discriminant analysis. LDA, QDA and logistic regression.
   - Robust statistics. Robust estimators of location and scale, M-estimators. Robust regression. Non-parametric tests.
   - Dimensionality reduction, task definition, manifold, intrinsic dimension. PCA and kernel PCA. Non-linear dimensionality reduction methods.
   - Clustering. The task formalization and its complexity. Clustering methods: k-means, EM GMM clustering, hierarchical clustering, density-based clustering. Spectral clustering.

5. **Principles of shape psychology. Essential composition and form principles.** BE4M39PTV

- Form-forming laws of shape psychology, their principles and applicability within interdisciplinary interaction (for example, in the designer / programmer relationship).
- Law of proximity, Law of similarity, Law of continuation / direction, Prägnanz, Law of good shape, Perception of figure and background, Constancy of size.
- Format and its interpretation within the form of shape psychology. The importance of the whole and its individual parts in favor of understanding and interpretation of composition (classical artwork-design, but also digital or multimedia content).
- The importance of color as an important part of the concept of designing a art / design, but also digital or multimedia content (contrast: luminosity, saturation, proportional, simultaneous, complementary).

6. **The methodology of software testing. Methods for test creation from the application model. Automated testing.** BE4M36ZKS (*Course web pages*)

- Describe and compare the V and W models of the software testing process. Explain static testing and its role in the W model. Describe individual methods of static testing.
- Explain the principle of Model Based Testing (MBT) and compare its advantages and disadvantages with manual testing approach. Give some examples of models that can be employed in MBT. How does MBT relate to test automation?
- Outline the main test automation principle and economics. What are possible levels at which tests can be automated? Give some examples of main approaches and technologies that can be used in software test automation.
- Explain the equivalence class and boundary values concepts and principle of the Combinatorial Interaction Testing. What is a combinatorial explosion effect, how to effectively reduce the input data combinations? Principle of pairwise (2-way) and N-way testing.
- Principle of path-based testing. Formal definition of system model and test coverage criteria (node/edge coverage, edge-pair coverage, prime-path coverage). How prioritization of process/workflow activities is modelled and handled in the generation of the test cases?

Software Engineering

1. **The methodology of software testing. Methods for test creation from the application model. Automated testing.** BE4M36ZKS (*Course web pages*)

- Describe and compare the V and W models of the software testing process. Explain static testing and its role in the W model. Describe individual methods of static testing.
- Explain the principle of Model Based Testing (MBT) and compare its advantages and disadvantages with manual testing approach. Give some examples of models that can be employed in MBT. How does MBT relate to test automation?
- Outline the main test automation principle and economics. What are possible levels at which tests can be automated? Give some examples of main approaches and technologies that can be used in software test automation.
- Explain the equivalence class and boundary values concepts and principle of the Combinatorial Interaction Testing. What is a combinatorial explosion effect, how to effectively reduce the input data combinations? Principle of pairwise (2-way) and N-way testing.
- Principle of path-based testing. Formal definition of system model and test coverage criteria (node/edge coverage, edge-pair coverage, prime-path coverage). How prioritization of process/workflow activities is modelled and handled in the generation of the test cases?

2. **Software architectures, their parameters and qualitative metrics. Architectural patterns, styles and standards.** BE4M36SWA (*Course web pages*)

- Describe Krutchen's 4+1 View Model of a software architecture. Explain how it captures the complete behavior of a developed software from multiple perspectives of the system. How is this model aligned with the UML models?
- What is a software architecture? Describe the importance of software architecture when developing a system. What are the software architecture design guidelines? What are the architectural styles? Give an example of an architectural style and describe it in detail.
- What is a design pattern? What problem are design patterns solving? What types of design patterns exist? Why is it important to know the design patterns? Are there design antipatterns?
- What is a microservice architecture? What are its advantages and disadvantages compared to a monolithic architecture. How is microservices development different from developing a monolithic application? Are there any methodologies or guidelines or best practices to follow when developing microservices? Obviously, there are patterns that can be applied in this area, are there any antipatterns that should be avoided?
- Software architects can choose one architecture over another. The choice may affect the quality of the final product. Can you tell if one architecture is better than another or if one architecture is bad while the other is not? How can you measure the quality of an architecture? Can you measure the quality from different perspectives?

3. **Properties of parallel and distributed algorithms. Communication operations for parallel algorithms. Parallel algorithms for linear algebra.** BE4M35PAG (*Course web pages*)

- Describe basic communication operations used in parallel algorithms. Show cost analysis of one-to-all broadcast, all-to-all-broadcast, scatter, and all-to-all personalized communication on a ring, mesh, and hypercube. Describe All-Reduce and Prefix-Sum operations and outline their usage.
- Describe performance metrics and scalability for parallel systems. How efficiency of a parallel algorithm depends on the problem size and the number of processors? Derive isoefficiency functions of a parallel algorithm for adding numbers (including communication between processors) and explain how it characterizes the algorithm.

- Explain and compare two parallel algorithms for matrix-vector multiplication. Describe a parallel algorithm for matrix-matrix multiplication and explain the idea of Cannon's algorithm. Discuss the principle and properties of the DNS algorithm used for matrix-matrix multiplication.
- Outline the principle of sorting networks and describe parallel bitonic sort, including its scalability. Explain parallel enumeration sort algorithm on PRAM model, including its scalability.
- Explain all steps of a parallel algorithm for finding connected components in a graph given by the adjacency matrix. Using an example, illustrate a parallel algorithm for finding a maximal independent set in a sparse graph.

4. **Effective algorithms and optimization methods. Data structures, synchronization and multithreaded programs.** BE4M36ESW (*Course web pages*)

- Java Virtual Machine, memory layout, frame, stack-oriented machine processing, ordinary object pointer, compressed ordinary object pointer. JVM bytecode, Just-in-time compiler, tired compilation, on-stack replacement, disassembler, decompiler. Global and local safe point, time to safe point. Automatic memory Management, generational hypothesis, garbage collectors. CPU and memory profiling, sampling and tracing approach, warm-up phase.
- Data races, CPU pipelining and superscalar architecture, memory barrier, volatile variable. Synchronization - thin, fat and biased locking, reentrant locks. Atomic operations based on compare-and-set instructions, atomic field updaters. Non-blocking algorithms, wait free algorithms, non-blocking stack (LIFO).
- Static and dynamic memory analysis, shallow and retained size, memory leak. Data Structures, Java primitives and objects, auto-boxing and unboxing, memory efficiency of complex data structures. Collection for performance, type specific collections, open addressing hashing, collision resolution schemes. Bloom filters, complexity, false positives, bloom filter extensions. Reference types - weak, soft, phantom.
- JVM object allocation, thread-local allocation buffers, object escape analysis, data locality, non-uniform memory allocation.
- Networking, OSI model, C10K problem. Blocking and non-blocking input/output, threading server, event-driven server. Event-based input/output approaches. Native buffers in JVM, channels and selectors.
- Synchronization in multi-threaded programs (atomic operations, mutex, semaphore, rw-lock, spinlock, RCU). When to use which mechanism? Performance bottlenecks of the mentioned mechanisms. Synchronization in "read-mostly workloads", advantages and disadvantages of different synchronization mechanisms.
- Cache-efficient data structures and algorithms (e.g., matrix multiplication). Principles of cache memories, different kinds of cache misses. Self-evicting code, false sharing – what is it and how deal with it?
- Profiling and optimizations of programs in compiled languages (e.g., C/C++). Hardware performance counters, profile-guided optimization. Basics of C/C++ compilers, AST, intermediate representation, high-level and low-level optimization passes.

5. **Big Data concept, basic principles of distributed data processing, types and properties of NoSQL databases.** BE4M36DS2 (*Course web pages*)

- Big Data (V characteristics, current trends), NoSQL databases (motivation, features). Scaling (vertical, horizontal, network fallacies, cluster). Distribution models (sharding, replication, master-slave and peer-to-peer architectures). CAP theorem (properties, ACID, BASE). Consistency (strong, eventual, read, write, quora). Performance tuning (Amdahl's law, Little's law, message cost model). Polyglot persistence.
- MapReduce (architecture, functions, data flow, execution, use cases). Hadoop (MapReduce, HDFS).

- XPath (path expressions, axes, node tests, predicates). XQuery (constructors, FLWOR, conditional, quantified and comparison expressions). SPARQL (subgraph matching, graph patterns, datasets, filters, solution modifiers, query forms).
- RiakKV (CRUD operations, links, link walking, convergent replicated data types, Search 2.0, vector clocks, Riak Ring, replica placement strategy). Redis (data types, operations, TTL). Cassandra (keyspaces, column families, CRUD operations). MongoDB (CRUD operations, update and query operators, projection, modifiers).
- Graph data structures (adjacency matrix, adjacency list, incidence matrix). Data locality (BFS layout, bandwidth minimization problem, Cuthill-McKee algorithm). Graph partitioning (1D partitioning, 2D partitioning). Neo4j (traversal framework, traversal description, traverser). Cypher (graph matching, read, write and general clauses).

6. **Security analysis of operating systems, development of secure software and web applications security. Analysis of cyberattacks and malware. Security of mobile devices.**

   BE4M36BSY (*Course web pages*)

- Managing a software project with a security as an objective, advantages and disadvantages of waterfall and ellipse model for this use-case, systematic identification of potential vulnerabilities, STRIDE, attack modelling (attack trees), ranking of vulnerabilities (ideal, DREAD).
- Timing and storage covert channels, Side channel attacks, Steganography.
- Discretionary access control(Access control list, Capabilities), Mandatory access control, Multi-level security, Biba model, Multi-lateral security, Role-based access control.
- Privilege escalation, security of operating systems, trusted computer base, reference monitor, complete mediation, needed mechanism for securing current OS, memory management, rings.
- Virtualization, virtual machine monitor, micro-kernels, general-purpose sandboxing, danger, Kernel namespaces, seccomp, Linux kernel capabilities.
- Access control model of web ecosystem, single-origin policy, preservations of integrity of data and code, sandboxing in web, content security policy.
- Network protocols, TCP, DNS, BGP, security of HTTPs, mechanism of certificates, security of certificate infrastructure.
- Firewalls, network intrusion detection, network intrusion prevention, thin client, intrusion deflection.
- Denial of service attack, reflection attacks, syn-cookies, detection and protection against DOS.

Computer Vision and Digital Image

1. **Basic data structures of computational geometry, methods of their construction, and representation.** BE4M39VG (*Course web pages*)

- Convex set, convex hull of a set (definitions). Convex hull (2D) representation. Its computation from a set of points: Graham's Algorithm, Jarvis' Algorithm of 'gift wrapping', Quick hull, and the Divide and Conquer method,. Computation of convex hull for a simple polygon. Computation and representation of convex hull in 3D. Complexity, dependency on the distribution of points and on the number of points on the convex hull.
- Point location in a polygon and in a planar subdivision (the method of slabs, the tree of monotonous chains, trapezoidal map). Representation of planar subdivision (DCEL).
- Voronoi Diagram (VD), Delaunay triangulation (DT), and their relation. VD computation from a set of points via the Incremental, Divide and Conquer, and Fortune's algorithm. DT computation by the Incremental method, edge legalization via edge-flip operation, maximization of angular sequence.

2. **Image representation for computer vision. Segmentation and image preprocessing methods.** BE4M33DZO (*Course web pages*)

- Image formation physics and its invertibility, consequences. Basics of optics for imaging. Optical aberrations. Image acquisition, sampling and quantization, image noise.
- Color images and their acquisition. Color metamer. CIE color space. Color triangle. Color camera, Bayer filter.
- Digital image, distance, neighbourhood, contiguity, region, boundary, distance transform.
- Image convolution, image filtering, linear and nonlinear (e.g., median) filter, edge (gradient), edgel.
- Image preprocessing: brightness and geometric transformations. Image interpolation. Histogram equalization. Linear and nonlinear noise filtration, edge and edgel detection. Canny detector. Interest points, Harris interest points.
- Fourier transform, definition in 1D, properties. Generalization to 2D. Image representation in frequency (Fourier) domain. Frequency filtering.
- Image segmentation methods: Histogram-based thresholding, segmentation by clustering, mean-shift-based segmentation, graph-based segmentation.
- Principal components analysis for images. Image entropy, redundancy, compression ratio. Signal to noise ratio. Lossy and lossless compression. Compression by coding. Compression of segmented images (chain code, RLE). Predictive compression (DPCM). JPEG and Discrete Cosine Transform (DCT).
- Mathematical morphology in lattice formalism. Dilation, erosion, opening, closing, hit-or-miss operations for both binary and gray scale images. Skeletonization.

3. **Object detection in images. Image matching and correspondence search.** BE4M33MPV (*Course web pages*)

- Object recognition and detection using deep neural networks. Network architecture, layer types, loss function.
- Image-to-image correspondence. Keypoint (i.e. distinguished region) detectors - Harris, Hessian, FAST MSERs. Local descriptors - SIFT, LBP, Hardnet. Establishing tentative correspondences.
- RANSAC algorithm, its application to estimation of geometric transformations and relations.
- Image retrieval. The bag of words method. Deep globally aggregated local descriptors.
- Tracking. The Kanade-Lucas-Tomasi (KLT) tracker. The Discrimination correlation tracker (DCT).

4. & 5. **Spatial geometry, image projection and perspective camera model for 3D reconstruction, virtual reality, visual odometry and SLAM. Algorithms for 3D geometric model reconstruction from images.** BE4M33GVG (*Course web pages*), BE4M33TDV (*Course web pages*)
   - Affine and projective plane and space. Representation of points, lines, planes, angles and distances and elementary operations with these entities.
   - Mathematical model of a perspective camera: projection matrix. Calibration of camera intrinsic parameters from known objects, camera resection from six points, exterior orientation from three points, and the corresponding algorithms.
   - Camera pair models: Homography. Epipolar geometry and epipolar constraint.
   - 3D scene reconstruction from images. Homography in P2 and the 4-point algorithm. Fundamental matrix and the 7-point algorithm. Essential matrix and the 5-point algorithm.
   - Computing camera motion and scene structure from image correspondences. Reprojection error and its Sampson approximation. Triangulation with Sampson correction. Bundle adjustment, gauge freedom and minimal representations.

6. **Minimizing empirical risk. Maximum likelihood estimation, EM algorithm. Deep networks and their training. Classical and deep neural networks and their learning.** BE4M33SSU (*Course web pages*)
   - Empirical risk minimization: Risk and empirical risk of a predictor, generalization bounds and Hoeffding inequality, statistically consistent learning algorithms, Vapnik-Chervonenkis-dimension of a hypothesis class, SVMs and Kernel SVMs.
   - Maximum likelihood estimators and the EM algorithm: maximum likelihood estimator, consistency of an estimator, EM algorithm as maximization of a lower bound of the likelihood, E-step and M-step as block-coordinate descent for the lower bound.
   - Deep networks, supervised learning of networks: neurons, network architectures, convolutional networks, backpropagation and layer types, parameter initialisation, stochastic gradient descent.

Data Science

1. **Statistical analysis, models and their assessment. Dimensionality reduction. Clustering.** BE4M36SAN (*Course web pages*)
   - Multiple linear regression. Describe the model and its assumptions. Treatment of qualitative independent variables, collinearity and outliers. Decide whether a model is useful. Overfitting, feature selection, model regularization.
   - Non-linear regression, polynomial regression, splines, local regression.
   - Discriminant analysis. LDA, QDA and logistic regression.
   - Robust statistics. Robust estimators of location and scale, M-estimators. Robust regression. Non-parametric tests.
   - Dimensionality reduction, task definition, manifold, intrinsic dimension. PCA and kernel PCA. Non-linear dimensionality reduction methods.
   - Clustering. The task formalization and its complexity. Clustering methods: k-means, EM GMM clustering, hierarchical clustering, density-based clustering. Spectral clustering.
2. **Scientific visualization methods. Information visualization methods.** BE4M39VIZ (*Course web pages*)
   - Characteristics of spatial data and spatial fields. Grids used to represent spatial fields and their relation with data enrichment. Visualization of scalar fields with color mapping and contouring. Direct volume rendering.
   - Visualization of vector fields with glyphs, stream objects, and line integral convolution (LIC).
   - Characteristics of abstract data. Visualization of tabular (or n-dimensional) data. Visualization of relational data (hierarchies, directed acyclic graphs, and undirected graphs).
   - Visualization of text. Visualization of a single document. Visualization of document corpus.
   - Visualization of time-varying data. Time primitives and relations between them. Time domain, its type, granularity and structure.
3. **Ontologies. Basic principles of ontological engineering, semantic web technologies, basic principles and technologies of linked data.** BE4M33OSW (*Course web pages*)
   - Characteristics of Description Logics. Basics of ALC, SHOIN(D) and SROIQ(D) description logics, computational complexity of key reasoning in these logics. Tableau algorithm for ALC. Blocking conditions in description logics.
   - Semantic Web stack, RDF, OWL, SPARQL, SWRL, tractability of rules in description logics (DL-safe rules).
   - Basics of RDF Graph model, blank node semantics. SPARQL query execution. Possible evaluation semantics of SPARQL and their differences.
   - Defining principles of linked data. Hash vs. 303 URIs, Five star linked open data model.
4. **Minimizing empirical risk. Maximum likelihood estimation, EM algorithm. Deep networks and their training. Classical and deep neural networks and their learning.** BE4M33SSU (*Course web pages*)
   - Empirical risk minimization: Risk and empirical risk of a predictor, generalization bounds and Hoeffding inequality, statistically consistent learning algorithms, Vapnik-Chervonenkis-dimension of a hypothesis class, SVMs and Kernel SVMs.
   - Maximum likelihood estimators and the EM algorithm: maximum likelihood estimator, consistency of an estimator, EM algorithm as maximization of a lower bound of the likelihood, E-step and M-step as block-coordinate descent for the lower bound.
   - Deep networks, supervised learning of networks: neurons, network architectures, convolutional networks, backpropagation and layer types, parameter initialisation, stochastic gradient descent.

5. **Learnability models: PAC and online. Learnability of conjunctions and disjunctions. Bayesian networks. Reinforcement learning.** BE4M36SMU (*Course web pages*)

- PAC and online learnability models: definition, efficient learnability. Comparison of the two models: differences in assumptions (such as i.i.d. observations), mutual relations (does one imply the other?). Vapnik-Chervonenkis dimension. Necessary and sufficient conditions for learnability.
- If one or the other concept class is learnable in one or the other learnability models, show an algorithm that learns it in that model. Are they also learnable efficiently? Learning other hypothesis classes by reduction to learning conjunctions or disjunctions.
- Bayesian networks: the notion of conditional independence and how BN defines it for a set of random variables. Computing conditional probabilities and the most probable state from a BN. Estimating BN parameters through maximum likelihood.
- Reinforcement learning: state utility, optimal policy, value iteration, direct utility estimation, adaptive dynamic programming, temporal difference learning, exploration vs. exploitation, Q-learning, and SARSA. Policy search.

6. **Big Data concept, basic principles of distributed data processing, types and properties of NoSQL databases.** BE4M36DS2 (*Course web pages*)

- Big Data (V characteristics, current trends), NoSQL databases (motivation, features). Scaling (vertical, horizontal, network fallacies, cluster). Distribution models (sharding, replication, master-slave and peer-to-peer architectures). CAP theorem (properties, ACID, BASE). Consistency (strong, eventual, read, write, quora). Performance tuning (Amdahl's law, Little's law, message cost model). Polyglot persistence.
- MapReduce (architecture, functions, data flow, execution, use cases). Hadoop (MapReduce, HDFS).
- XPath (path expressions, axes, node tests, predicates). XQuery (constructors, FLWOR, conditional, quantified and comparison expressions). SPARQL (subgraph matching, graph patterns, datasets, filters, solution modifiers, query forms).
- RiakKV (CRUD operations, links, link walking, convergent replicated data types, Search 2.0, vector clocks, Riak Ring, replica placement strategy). Redis (data types, operations, TTL). Cassandra (keyspaces, column families, CRUD operations). MongoDB (CRUD operations, update and query operators, projection, modifiers).
- Graph data structures (adjacency matrix, adjacency list, incidence matrix). Data locality (BFS layout, bandwidth minimization problem, Cuthill-McKee algorithm). Graph partitioning (1D partitioning, 2D partitioning). Neo4j (traversal framework, traversal description, traverser). Cypher (graph matching, read, write and general clauses).

Cyber Security

1. **Statistical analysis, models and their assessment. Dimensionality reduction. Clustering.** BE4M36SAN (*Course web pages*)

- Multiple linear regression. Describe the model and its assumptions. Treatment of qualitative independent variables, collinearity and outliers. Decide whether a model is useful. Overfitting, feature selection, model regularization.
- Non-linear regression, polynomial regression, splines, local regression.
- Discriminant analysis. LDA, QDA and logistic regression.
- Robust statistics. Robust estimators of location and scale, M-estimators. Robust regression. Non-parametric tests.
- Dimensionality reduction, task definition, manifold, intrinsic dimension. PCA and kernel PCA. Non-linear dimensionality reduction methods.
- Clustering. The task formalization and its complexity. Clustering methods: k-means, EM GMM clustering, hierarchical clustering, density-based clustering. Spectral clustering.

2. **The methodology of software testing. Methods for test creation from the application model. Automated testing.** BE4M36ZKS (*Course web pages*)

- Describe and compare the V and W models of the software testing process. Explain static testing and its role in the W model. Describe individual methods of static testing.
- Explain the principle of Model Based Testing (MBT) and compare its advantages and disadvantages with manual testing approach. Give some examples of models that can be employed in MBT. How does MBT relate to test automation?
- Outline the main test automation principle and economics. What are possible levels at which tests can be automated? Give some examples of main approaches and technologies that can be used in software test automation.
- Explain the equivalence class and boundary values concepts and principle of the Combinatorial Interaction Testing. What is a combinatorial explosion effect, how to effectively reduce the input data combinations? Principle of pairwise (2-way) and N-way testing.
- Principle of path-based testing. Formal definition of system model and test coverage criteria (node/edge coverage, edge-pair coverage, prime-path coverage). How prioritization of process/workflow activities is modelled and handled in the generation of the test cases?

3. **Security analysis of operating systems, development of secure software and web applications security. Analysis of cyberattacks and malware. Security of mobile devices.**

BE4M36BSY (*Course web pages*)

- Managing a software project with a security as an objective, advantages and disadvantages of waterfall and ellipse model for this use-case, systematic identification of potential vulnerabilities, STRIDE, attack modelling (attack trees), ranking of vulnerabilities (ideal, DREAD).
- Timing and storage covert channels, Side channel attacks, Steganography.
- Discretionary access control(Access control list, Capabilities), Mandatory access control, Multi-level security, Biba model, Multi-lateral security, Role-based access control.
- Privilege escalation, security of operating systems, trusted computer base, reference monitor, complete mediation, needed mechanism for securing current OS, memory management, rings.
- Virtualization, virtual machine monitor, micro-kernels, general-purpose sandboxing, danger, Kernel namespaces, seccomp, Linux kernel capabilities.
- Access control model of web ecosystem, single-origin policy, preservations of integrity of data and code, sandboxing in web, content security policy.

- Network protocols, TCP, DNS, BGP, security of HTTPs, mechanism of certificates, security of certificate infrastructure.
- Firewalls, network intrusion detection, network intrusion prevention, thin client, intrusion deflection.
- Denial of service attack, reflection attacks, syn-cookies, detection and protection against DOS.

4. **Symmetric and asymmetric cryptography. Basic cryptosystems (RSA, El-Gamal). Number factorisation. Hashing algorithms.** BE4M01MKR (*Course web pages*)

- RSA cryptosystem and number theory behind it (Euler theorem, Repeated squaring algorithm, the Chinese remainder theorem). Attacks on RSA cryptosystem.
- Number factorisation problem and its use in cryptography. A subexponential algorithm for factorisation, Quadratic sieve algorithm - basic ideas of their principles and time complexity.
- Primes and their properties, generating random primes. Probabilistic primality testing, Fermat test, Miller-Rabin test - basic principles and error probability.
- Diffie-Hellman key exchange, El-Gamal cryptosystem and group theory behind it (cyclic groups, finding a generator, which groups Z_n^* are cyclic?).
- Discrete logarithm problem and its use in cryptography. Baby step-giant step algorithm, Pohling-Hellman algorithm and a subexponential algorithm for discrete logarithm - basic ideas of their principles and time complexity.
- Elliptic Curve Cryptography - Diffie-Hellman key exchange and El-Gamal cryptosystem using an elliptic curve group. Elliptic curve discrete logarithm problem and Baby step-giant step algorithm.

5. **Network routing principles. Network transport protocols. Software defined networks. Network function virtualization.** A0M32PST (*Course web pages*)

- Explain function of Border Gateway Protocol. What Are Different Bgp message types and their meanings. Explain Bgp path attributes and their influence on best route selection.
- What are Benefits of using MPLS? Describe Push, Swap, Pop operations in Mpls network and also explain Penultimate Hop Popping. Describe MPLS network architecture for virtual private networks. What is the difference between RD and RT. How MPLS labels are discovered and assigned in the MPLS networks.
- Describe a role of IP multicast. What Are The Protocols Used In IP Multicast. Explain a role of IGMP in IP multicast networks. Explain two versions of multicast tree and the difference between PIM sparse and dense modes.
- How does Ipv6 solve the problem of Ipv4 address exhaustion? Explain format of IPv6 address and its parts. Explain a function of stateless IPv6 address autoconfiguration (SLAAC) and protocol for discovering network neighbours.

6. **Principles of secure system design. Design and analysis of secure communication protocols, e.g., TLS, mobile telephony and others. Distributed system security.** BE4M36KBE

- Cipher type & modes - stream, block, ECB, CBC, CTR, GCM, etc. - their strengths, weaknesses, implications on confidentiality and integrity protection.
- Secure communication channel set-up using combination of asymmetric and symmetric cryptography primitives: authenticating the other party, protecting confidentiality and integrity.
- Authentication and authorisation using symmetric cryptography primitives - Needham-Scroeder, Kerberos, authentication protocols and systems in mobile telephony: GSM, 3G, LTE authentication principles and protocols.

- TLS protocol, its motivation, evolution, weaknesses and attacks, differences between the version 1.2 and 1.3.
- Blockchain and Bitcoin principles - Private key, address, transaction, block, blockchain, mining, transaction verification.

Computer Engineering

1. **Design and implementation of in-chip integrated systems, application specific systems.** BE4M34ISC (*Course web pages*)

- Main features and economical aspects of the Application specific integrated circuits systems: full custom design, gate array, standard cells, programmable array logic;
- Design principles of mix-signal integrated circuits, purpose of hierarchical design, digital and analogue block interface, CAD design tools for automatic circuit generation; functional and static time analysis, formal verification; Verilog-A, Verilog-AMS, VHDL-A.
- Front end design - functional specification, RTL, logic synthesis, Gate-level netlist, behavioral stimulus extraction.
- Back End design - specification of Design Kit, Floorplanning, place and route, layout, parasitic extraction, layout versus schema check (LVS).
- Tape out and IC fabrication process, integrated systems verification, scaling and design mapping to different technologies.

2. **Advanced architectures of processors, memory and peripheral circuits and multiprocessor computers.** BE4M35PAP (*Course web pages*)

- Superscalar techniques used in nodes of multiprocessor systems, data flow inside the processor, Tomasulo algorithm and its deficiencies, precise exceptions support, architectural state, register renaming, reservation station, reorder buffer, instruction fetch, decode, dispatch, issue, execute, finish, complete, reorder, branch prediction, store forwarding, hit under miss.
- Relation between memory coherency and consistency, their implementation on systems with shared bus and when multiple rings topologies are used, MESI, MOESI, home directory.
- Rules for execution synchronization and data exchange in multiprocessor systems, mutex implementation, relation to consistency models and mechanisms to achieve expected algorithms behavior on systems with relaxed consistency models (PRAM, PSO, TSO, PC, barrier instructions).
- SMP and NUMA nodes interconnections networks, conflicts and rearrangeable networks, Beneš network.
- Parallel computations on multiprocessor systems, OpenMP on NUMA and MPI on distributed memory systems, their combinations.

3. **I/O and network interfaces of computer and embedded systems, hardware and software implementation.** BE4M38KRP (*Course web pages*)

- USB I/O subsystem, structure and functionality of elements, protocol stack,transfer - transaction - packet hierarchy, transfer types and pipes, bandwidth allocation principles, enumeration process and PnP, descriptor hierarchy, USB device implementation.
- PCI Express (PCI) I/O subsystems, basic differences and commons of PCI and PCIe, protocol stack, transaction types, packet routing principles, quality of service support, PnP and enumeration process.
- Ethernet based networking, VLAN, precision time protocol (PTP), stream reservation protocol (SRP), time sensitive networks (TSN).
- In-vehicle networking, Controller Area Network (CAN, CAN-FD), Local Interconnect Network (LIN), FlexRay, data-link layer algorithms, physical topology constraints and relation to system design.

4. **ARM based microcontrollers and signal processors; their functionality. Design and implementation of embedded systems for typical application areas.** BE4M38AVS (*Course web pages*)

- Typical architecture and main features of ARM based microcontrollers. AMBA. I/O pin configuration. Common used peripheral circuits (I/O ports, timers, DMA controllers, NVIC controller, JTAG, SWD, A/D converters, D/A converters, SPI controllers, I2C controllers, UART, FLASH and SRAM memory).
- Typical architecture and main features of digital signal processors (DSP). Common used peripheral circuits. Special computational units and their features (ALU, MAC, SHIFT BARREL register, DAG).
- Digital signal processing: signal spectrum analysis (DFT, IDFT), correlation functions and their typical use, digital filters (FIR, IIR), signal interpolation, signal decimation.
- Types of A/D converters. Sampling theorem. Anti-aliasing filter (AAF). Direct digital synthesis (DDS).
- User controls interfacing to microcontrollers (buttons, rotary encoders, graphic LCD, audio codecs, power switches, relays, contactors). Motion control (brush DC motor, stepper motor and brushless DC motor control).

5. **Properties of parallel and distributed algorithms. Communication operations for parallel algorithms. Parallel algorithms for linear algebra.** BE4M35PAG (*Course web pages*)

- Describe basic communication operations used in parallel algorithms. Show cost analysis of one-to-all broadcast, all-to-all-broadcast, scatter, and all-to-all personalized communication on a ring, mesh, and hypercube. Describe All-Reduce and Prefix-Sum operations and outline their usage.
- Describe performance metrics and scalability for parallel systems. How efficiency of a parallel algorithm depends on the problem size and the number of processors? Derive isoefficiency functions of a parallel algorithm for adding numbers (including communication between processors) and explain how it characterizes the algorithm.
- Explain and compare two parallel algorithms for matrix-vector multiplication. Describe a parallel algorithm for matrix-matrix multiplication and explain the idea of Cannon's algorithm. Discuss the principle and properties of the DNS algorithm used for matrix-matrix multiplication.
- Outline the principle of sorting networks and describe parallel bitonic sort, including its scalability. Explain parallel enumeration sort algorithm on PRAM model, including its scalability.
- Explain all steps of a parallel algorithm for finding connected components in a graph given by the adjacency matrix. Using an example, illustrate a parallel algorithm for finding a maximal independent set in a sparse graph.

6. **Effective algorithms and optimization methods. Data structures, synchronization and multithreaded programs.** BE4M36ESW (*Course web pages*)

- Java Virtual Machine, memory layout, frame, stack-oriented machine processing, ordinary object pointer, compressed ordinary object pointer. JVM bytecode, Just-in-time compiler, tired compilation, on-stack replacement, disassembler, decompiler. Global and local safe point, time to safe point. Automatic memory Management, generational hypothesis, garbage collectors. CPU and memory profiling, sampling and tracing approach, warm-up phase.
- Data races, CPU pipelining and superscalar architecture, memory barrier, volatile variable. Synchronization - thin, fat and biased locking, reentrant locks. Atomic operations based on compare-and-set instructions, atomic field updaters. Non-blocking algorithms, wait free algorithms, non-blocking stack (LIFO).
- Static and dynamic memory analysis, shallow and retained size, memory leak. Data Structures, Java primitives and objects, auto-boxing and unboxing, memory efficiency of complex data structures. Collection for performance, type specific collections, open addressing hashing, collision resolution schemes. Bloom filters, complexity, false positives, bloom filter extensions. Reference types - weak, soft, phantom.

- JVM object allocation, thread-local allocation buffers, object escape analysis, data locality, non-uniform memory allocation.
- Networking, OSI model, C10K problem. Blocking and non-blocking input/output, threading server, event-driven server. Event-based input/output approaches. Native buffers in JVM, channels and selectors.
- Synchronization in multi-threaded programs (atomic operations, mutex, semaphore, rw-lock, spinlock, RCU). When to use which mechanism? Performance bottlenecks of the mentioned mechanisms. Synchronization in "read-mostly workloads", advantages and disadvantages of different synchronization mechanisms.
- Cache-efficient data structures and algorithms (e.g., matrix multiplication). Principles of cache memories, different kinds of cache misses. Self-evicting code, false sharing – what is it and how deal with it?
- Profiling and optimizations of programs in compiled languages (e.g., C/C++). Hardware performance counters, profile-guided optimization. Basics of C/C++ compilers, AST, intermediate representation, high-level and low-level optimization passes.

Bioinformatics

1. **Chemical composition of living matter, experimental models and methods, genetic code.** BE4M36MBG

- Chemical composition of living matter, experimental models and methods, genetic code.
- Chemical properties of water and its importance for life. Main and minor biogenic elements. Comparison of importance of covalent chemical bonds and weak interactions in a living cell.
- Building blocks of biopolymers. Proportion of different biopolymers and their building blocks in the cell. Chemical composition, primary, secondary and tertiary structures of proteins and nucleic acids.
- Model organisms in molecular biology. Characteristics of the ideal model organisms. Sequencing projects and what they brought to us? What does it mean in reality if some genome is claimed to be sequenced. What are the pitfalls of genome annotation. Differences between annotation of the bacterial and eukaryotic genomes. ENCODE project.
- Basic principles of methods of high-throughput genome, transcriptome and/or proteome analyses.
- Genetic code - structure, properties, significance and evolution. How cells use the genetic code in protein synthesis.

2. **Modeling and analysis of biological sequences.** BE4M36BIN (*Course web pages*)

- The basic overview of DNA sequencing approaches, sequence assembly (problem definition, algorithms, common assembly difficulties).
- Sequence alignment (the connection between similarity and homology, alignment scoring, exact and heuristic solutions, pairwise alignment vs multiple sequence alignment).
- Distance-based, parsimony-based and probabilistic approaches to phylogenetic tree construction (principles, assumptions, the differences between the approaches).
- Markov models of genomic sequences. Markov chains and hidden Markov models, their structure, learning and genomic applications.
- Gene expression and its analysis. Differential gene expression and statistical models for its discovery.
- Structure prediction. Levels of protein description, the role of higher structures of RNA and proteins in computational biology, structure prediction.

3. **Image representation for computer vision. Segmentation and image preprocessing methods.** BE4M33DZO (*Course web pages*)

- Image formation physics and its invertibility, consequences. Basics of optics for imaging. Optical aberrations. Image acquisition, sampling and quantization, image noise.
- Color images and their acquisition. Color metamer. CIE color space. Color triangle. Color camera, Bayer filter.
- Digital image, distance, neighbourhood, contiguity, region, boundary, distance transform.
- Image convolution, image filtering, linear and nonlinear (e.g., median) filter, edge (gradient), edgel.
- Image preprocessing: brightness and geometric transformations. Image interpolation. Histogram equalization. Linear and nonlinear noise filtration, edge and edgel detection. Canny detector. Interest points, Harris interest points.
- Fourier transform, definition in 1D, properties. Generalization to 2D. Image representation in frequency (Fourier) domain. Frequency filtering.
- Image segmentation methods: Histogram-based thresholding, segmentation by clustering, mean-shift-based segmentation, graph-based segmentation.

- Principal components analysis for images. Image entropy, redundancy, compression ratio. Signal to noise ratio. Lossy and lossless compression. Compression by coding. Compression of segmented images (chain code, RLE). Predictive compression (DPCM). JPEG and Discrete Cosine Transform (DCT).
- Mathematical morphology in lattice formalism. Dilation, erosion, opening, closing, hit-or-miss operations for both binary and gray scale images. Skeletonization.

4. **Statistical analysis, models and their assessment. Dimensionality reduction. Clustering.** BE4M36SAN (*Course web pages*)
- Multiple linear regression. Describe the model and its assumptions. Treatment of qualitative independent variables, collinearity and outliers. Decide whether a model is useful. Overfitting, feature selection, model regularization.
- Non-linear regression, polynomial regression, splines, local regression.
- Discriminant analysis. LDA, QDA and logistic regression.
- Robust statistics. Robust estimators of location and scale, M-estimators. Robust regression. Non-parametric tests.
- Dimensionality reduction, task definition, manifold, intrinsic dimension. PCA and kernel PCA. Non-linear dimensionality reduction methods.
- Clustering. The task formalization and its complexity. Clustering methods: k-means, EM GMM clustering, hierarchical clustering, density-based clustering. Spectral clustering.

5. **Learnability models: PAC and online. Learnability of conjunctions and disjunctions. Bayesian networks. Reinforcement learning.** BE4M36SMU (*Course web pages*)
- PAC and online learnability models: definition, efficient learnability. Comparison of the two models: differences in assumptions (such as i.i.d. observations), mutual relations (does one imply the other?). Vapnik-Chervonenkis dimension. Necessary and sufficient conditions for learnability.
- If one or the other concept class is learnable in one or the other learnability models, show an algorithm that learns it in that model. Are they also learnable efficiently? Learning other hypothesis classes by reduction to learning conjunctions or disjunctions.
- Bayesian networks: the notion of conditional independence and how BN defines it for a set of random variables. Computing conditional probabilities and the most probable state from a BN. Estimating BN parameters through maximum likelihood.
- Reinforcement learning: state utility, optimal policy, value iteration, direct utility estimation, adaptive dynamic programming, temporal difference learning, exploration vs. exploitation, Q-learning, and SARSA. Policy search.

6. **Minimizing empirical risk. Maximum likelihood estimation, EM algorithm. Deep networks and their training. Classical and deep neural networks and their learning.** BE4M33SSU (*Course web pages*)
- Empirical risk minimization: Risk and empirical risk of a predictor, generalization bounds and Hoeffding inequality, statistically consistent learning algorithms, Vapnik-Chervonenkis-dimension of a hypothesis class, SVMs and Kernel SVMs.
- Maximum likelihood estimators and the EM algorithm: maximum likelihood estimator, consistency of an estimator, EM algorithm as maximization of a lower bound of the likelihood, E-step and M-step as block-coordinate descent for the lower bound.
- Deep networks, supervised learning of networks: neurons, network architectures, convolutional networks, backpropagation and layer types, parameter initialisation, stochastic gradient descent.