Rémy Grünblatt

Student in computer science

101 avenue Debourg
69007 Lyon
+33 6 51 74 06 13
remy.grunblatt@ens-lyon.fr

Education

2016-2017 **Master 2 Informatique fondamentale**, *ENS de Lyon*, Lyon, France.
Second year of Master in Computer Science.

2015-2016 **Master 1 Informatique fondamentale**, *ENS de Lyon*, Lyon, France.
First year of Master in Computer Science.

2014-2015 **Licence 3 Informatique fondamentale**, *ENS de Lyon*, Lyon, France.
BSc in Computer Science.

2011 – 2014 **Classe préparatoire aux Grandes Écoles, MPSI, MP**, *Lycée Kléber*, Strasbourg, France.
Preparation for the national competitive examination for admission to the French "Grandes Écoles". Mathematics, Physics and Computer Science.

2011 **Baccalauréat Scientifique**, *Lycée Sainte-Marie*, Belfort, France.
High-school diploma with scientific major. Mathematics, physics and biology.

Internships – Work experience

February 2017, 5 months **Research internship**, ENS de Lyon, Lyon, France, Ongoing.
A New Paradigm for Data Management in Cloud Environment: a data centric approach.

Summer 2016, 3 months **Research internship**, CETIC, Charleroi, Belgium, supervised by **Sébastien Dupont** and **Stéphane Mouton**.
Design and development of a distributed key-value store resilient to byzantine faults, using blockchain inspired techniques.

Summer 2015, 6 weeks **Research internship**, IRISA, Lannion, France, in the CAIRN research team, supervised by **Arnaud Tisserand**.
Hardware-Software co-design for the Zynq Soc (Zedboard): Design and development of an OpenSSL cryptographic offloading module for the elliptic-curve cryptography processor Pavois.

Skills

Programming C++, Python, C, Ocaml, Bash, R

Libraries Scapy, MPI, ZeroMQ, Protobuf

Tools GIT, Linux / Unix systems, LaTeX

Misc System administration, software defined radio, electronic, soldering, ...

Languages

French Mother tongue

English Fluent

Miscellaneous

2015 Google Hash Code 2015 programming contest: Ranked #6 (as a part of a 4 members team) over more than 2000 teams.

2014 – 2017 President of the student "IT" association of the ENS de Lyon, AliENS. The association gives introduction courses for students on programming languages, development tools, privacy, security, and provide services and technical support to (mainly) students. The association currently aims to provide wireless network in the student dorms.

2017 Member of the student representative council (Élus étudiants), member of the "Council of studies and student life" (CEVE, *Conseil des Etudes et de la Vie Etudiante*) and member of the steering committee for the information system security of the ENS de Lyon (CoPil SSI, *Comité de Pilotage Sécurité des Systèmes d'Information*).

Interests

- Free and Open-Source Softwares
- Science-Fiction
- Volley
- Information Security
- Cinema
- Bande Dessinées

Computer science courses followed in 2016-2017

First Semester **Network Information Theory**, 24 hours.

Fundamental limits of data transmission in several canonical multi-user channels: multiple access channels; broadcast channels; interference channels; relay channels. Achievability, converse and capacity region; application to simultaneous information and energy transmission.

Complex Networks, 24 hours.

Introduction to complex networks (ErdsRényi, Small-World and BarabásiAlbert model) scale-free, small-world), structural phase-transitions, node and link centrality measures, network modularity and community detection methods, temporal, multiplex and spatial networks, and network construction practical analysis.

Resilient and Energy-Aware Scheduling Algorithms, 24 hours.

Introduction to resilience strategies and algorithms for exascale: checkpointing, failure prediction, silent errors; quantified analysis and comparisons of strategies; introduction to energy-aware scheduling.

Network Algorithms for Molecular Biology, 24 hours.

Introduction to graph algorithms used in molecular biology; reconciliation of gene trees with phylogenetic networks.

Program Analysis, Safety Program Verification, 24 hours.

Introduction to model-checking using SAT and SMT-based program analysis. Lectures on abstract interpretation and their applications to verification, termination, and compilation.

Advanced Compilers: Automatic Parallelization and High-level Synthesis, 24 hours.

Introduction to static parallelization and to the polyhedral model; introduction to high-level synthesis for FPGA, sparse data-flow analysis and data-movement complexity analysis.

11 - 15 january 2016 **Research School on "Molecular programming: from theory to wet lab nano-scale computation", by Ludovic Bellon, Cendrine Moskalenko, Yannick Rondelez, Nicolas Schabanel and Damien Woods**, *ENS de Lyon*, Lyon, France, 24 hours of lecture.

Introduction to molecular programming using DNA: complexity theory, computation models and algorithms. Experimental sessions in wet-lab: discrete computation using DNA tiles, analog computation using DNA strands to implement numerical partial differential equations

7 - 12 december 2015 **Resarch School on "Scientific Methodology and Performance Evaluation for Computer Scientists" by Arnaud Legrand**, *ENS de Lyon*, Lyon, France, 24 hours of lecture.

Fundamental basis for sound scientific methodology of performance evaluation of computer systems; methodological aspects of measurement and statistics; reproducible research, data visualisation and introduction to probabilities/statistics.

Computer science courses followed in 2015-2016

First semester **Performance Evaluation and Networks**, 56 hours.

Reminder in probabilities; discrete and continuous time Markov chains and ergodicity, Poisson processes, queuing theory and networks analysis. Introduction to the NS3 simulator. Introduction to statistics.

Optimisation and Approximation, 56 hours.

Introduction to linear programming, polytopes representation of problems, and simplex and ellipsoid algorithms. Approximation algorithms, using deterministic or randomized rounding.

Parallel and Distributed Systems, 56 hours.

Sorting networks, PRAM models, algorithms on rings and grids of processors, scheduling, introduction to automatic parallelization distributed systems and wave algorithms. Implementation of a distributed algorithm to compute stencil operations (wave propagation) using MPI.

Compilers and Program Analysis, 56 hours.

Notions of compilation: grammars, intermediary representations, register allocation, partial implementation of a C compiler; Notinos of program analysis: abstract interpretation, Hoare logic, typing and functional languages.

11 - 15 january 2016 **Research School on "Data Mining: Statistical Modeling and Learning from Data", by Ciro Cattuto, Laetitia Gauvin and André Panisson, ISI Torino**, *ENS de Lyon*, Lyon, France, 24 hours of lecture.

Introduction to machine learning (learning problem, approximation, generalization, learning curve); linear models (linear regression, logistic regression, lasso), non-linear models (SVM, naive Byes, decision tree, random forests), unsupervised learning (SVD, NMF, k-means, text-analysis) - http://andre.panisson.com/datamining/

7 - 12 december 2015 **Resarch School on "Probabilistic Techniques and Models in Computer Science", by Stefan Kiefer, Joel Ouaknine and James Worrel, University of Oxford**, *ENS de Lyon*, Lyon, France, 24 hours of lecture.

Introduction to randomized algorithms: decision problems (space-bounded intreractive protocols, reachability and threshold problems for markov chains), stochastic processes (markov chains, martingales, optionnal stopping theorem), and learning theory (Vapnik-Chervonenkis dimension)

Second semester **Computer Algebra**, 40 hours.

Arithmetics algorithms (mostly polynomial): representation of polynomials, linear operation, multiplication and division using schoolbook, Karatsuba, Toom-Cook, Fast Fourier Transform methods. GCD, CRT and their fast alternative. Linear algebra and polynomial system solving over a field, fast matrix operations.

Distributed systems, 40 hours.

Distributed systems modelization and communication protocols, wave, traversal, leader election and mutual exclusion algorithms. Fault Tolerance models and algorithms, self stabilization and clock synchronization algorithms. Practical sessions in Erlang.

Data Bases and Data Mining, 40 hours.

Introduction to databases, relational languages, `SQL`, functional dependencies, data exchange, data mining (itemset mining, sequence mining) and data learning.

Computational Geometry and Digital Images, 40 hours.

Study of the representation of images and basic shapes; basic of image processing, digital geometry for shape analysis, data structures.

Whole year **Integrated project: Design of an easy to use distributed agent-based simulator**, 64 hours.

Implementation in `C++`, using mainly `MPI` for parallelism and distribution of the work, `Clang` for source-to-cource compilation from a high level homemade language to `C++` and `Qt` for the graphical user interface.

Computer science courses followed in 2014-2015

First semester **Algorithmics 1**, 64 hours.

Classical algorithm design paradigms (divide-and-conquer, dynamic programming, greedy algorithms), complexity (space, time, average and worst case, amortized analysis), and NP-completeness (reductions and approximations).

Architecture of Computers, 64 hours.

Technological and theoretical overview of the architecture of computers. Creation of a simulation of a computer from logic gates in the LogiSim tool.

Computer Science Foundations and Computability, 64 hours.

Finite automata and regular languages, pushdown automata, grammars and context-free languages. Turing Machines, undecidable problems, halting problem, and Rice's theorem.

Programming Languages Theory, 64 hours.

Operational and axiomatic semantics of small toy languages, rewriting and typing. Introduction to the Coq Proof Assistant.

Programming Project in Java, 64 hours.

Introduction to the Java language: implementation of the k-means algorithm, construction of Voronoi diagrams and spectral clustering.

Programming Project in Ocaml, 64 hours.

Introduction to the Ocaml language with a small toy project.

Second semester

Algorithmics 2, 64 hours.

Data structures: trees, heaps, graphs, union-find, hash tables; graphs algorithmics: distances, spanning trees, flows; words algorithmics: KMP.

Probabilities, 64 hours.

Theoretical courses about probabilities. Basic notions: probability distributions, random variables, classical inequalities and convergence theorems; discrete Markov chains.

Preparation for the ACM-ICPC programming contest, 64 hours.

Practical illustration of the Algorithmics courses based on solving ACM-ICPC regional problems (competitive programming).

Systems and networks, 64 hours.

Introduction to the architecture of operating systems and networks. OS structure, threads, synchronization algorithm, process scheduling and memory organization. Introduction to networks, TCP/IP, UDP. Implementation in C of a simple BitTorrent client.

SAT solver project, 64 hours.

Implementation of a SAT/SMT solver by groups of two students. Implementation of the DPLL algorithm in `OCaml` with watched literals and clause learning, Tseitin transformation, and an on-line SMT solver using DPLL(T) for the equality and linear arithmetics theories.