Online Schedule at https://docs.google.com/spreadsheets/d/1IBtE0tcp5sAV-e4dAVq6qt1OSzAXWUWkQeEPxmF7AEU/edit#gid=79405663

Syllabus for Algorithms and Data Structures 2, Fall 2019-20, program Applied Data Analysis, LSE and HSE double diploma program

Instructor: Boris Goldengorin, goldengorin@gmail.com

1. Course Description

a. Title of Course: Algorithms and Data Structures 2

b. **Pre-requisites**: Discrete Mathematics 1, Algorithms and Data Structures 1, Microsoft Excel or any general purpose solver (software), e.g. MATLAB (https://www.mccormick.northwestern.edu/documents/students/undergraduate/introduction-to-matlab.pdf),

Xpress (https://www.msi-jp.com/xpress/learning/square/18-nonlinear.pdf),

CPLEX (https://www.ibm.com/support/knowledgecenter/SSSA5P_12.8.0/ilog.odms.studio.help/pdf/gscplex.pdf),

etc.

c. **Course Type**: compulsory

d. **Abstract:** The objective of this course is to make students familiar with polynomially solvable (tractable) and intractable (NP-complete and NP-hard) problems including well known problems in combinatorial optimization and computer science (e.g. the traveling salesman problem and its polynomially solvable special cases; generalized p-median problem and its applications in data analysis; max-clique, basic models of integer linear programming models (discrete optimization models) and enumeration (branch-and-bound type) algorithms for solving general mixed integer linear programming problems as well as its special cases (minimum spanning tree and its variations, linear assignment, and traveling salesman problem). The algorithmic line of this course will be represented by introduction to the maximization of submodular set functions (discrete analog of convex functions), and their applications to data analysis problems. We conclude this course with an overview of time and space complexities for polynomially solvable and NP-hard problems.

2. Learning Objectives

Students who complete this course successfully will learn or acquire:

- Basic concepts of computational complexities for problems and algorithms (both exact and heuristic), which are necessary for further learning computational methods, algorithms for big data, algorithms and computational complexity, machine learning, operations research, game theory, and combinatorial optimization.
- Skills in using algorithms and data structures 2 models and methods to formalize and solve applied problems

3. Learning Outcomes

- Able to find mathematical structures defined on graphs and Booleans (Hasse diagrams)
- Learn basic notions of algorithms and data structures to solve intractable problems in computer science
- Basic skills to recognize and prove the time and space complexities for exact and heuristic algorithms

4. Course Outcomes:

Outcomes of the course are related to the program outcomes of applied data analysis within double diploma program of LSE and NRU HSE. Upon the completion of the course students will be able to recognize polynomially solvable and NP-hard (NP-complete) problems including efficient solution methods, exact, approximation algorithms, and useful data structures.

4.1. **Examples of polynomially solvable problems**: the Minimum Spanning Tree and Its variations; Linear Assignment Problem (LAP) and Patterns within LAP; Max-Flow-Min-Cut; Polynomially Solvable Special Cases of NP-hard problems: the Symmetric and Asymmetric versions of Traveling Salesman Problem; Generalized p-Median Problem and Its Applications in Data Analysis; Single Machine Scheduling Problems and Their Applications in Computer Science; Submodular Set Functions and Their Applications in Data Analysis.

4.2. **Branch and bound algorithms for solving NP-hard Problems**: Branch and Bound, Data Correcting, Branch and Cut, Cutting Planes, Branch and Cut and Price, Parallel Branch and Bound, Tolerance Based Branch and Bound, Climer and Zhang's cut-and-solve algorithms.

4.3. **Introduction to computational complexity**: The theory of NP-Completeness; Relationship between P and NP classes; Polynomial transformations from a language (problem) to another language (problem); Cook, Karp, and Levin's Theorem.; Algorithm for proving NP- Completeness; Six basic NP-Complete problems: 3-Satisfiability, 3-Dimensional Matching, Vertex Cover, Clique, Hamiltonian cycle (circuit).

5. Learning Outcomes

- Able to find mathematical structures in the object you work with
- Learn basic math notions that are needed to work in broad areas of applied data analysis
- Basic math skills for learning advanced courses in applied data analysis and computer science.

6. Course outcomes:

Outcomes of the course are related to the program outcomes of applied data analysis within double diploma program of LSE and NRU HSE. Upon the completion of the course students will be able to model, solve and make conclusions based on the following notions, problems and solution methods including exact and heuristic algorithms.

1. The Minimum Spanning Tree (Kruskal and Prim's Algorithms) 1-Tree, and Linear Assignment Problems (Hungarian Algorithm).
2. The Traveling Salesman Problem (TSP): Symmetric and Asymmetric Cases of TSP.
3. The Branch and Bound Algorithms for the Symmetric TSP (STSP).
4. The Branch and Bound Algorithms for the Asymmetric TSP (ATSP).
5. Upper, Lower, and Bottleneck Tolerances in Combinatorial Optimization.
6. Upper Tolerances Based Algorithm for the ATSP.
7. Lower Tolerances Based Algorithm for the ATSP.
8. Data Correcting Approach for Real-Valued Functions.
9. Data Correcting Approach for ATSP.
10. Heuristics for solving the TSP: greedy (nearest neighbor), tolerance based greedy, nearest insertion, 2, 3-opt.
11. Introduction to Submodular Functions: Local, Global Maxima on Hasse Diagram and Their Components.
12. Supermodular Functions Applied to the Simple Plant (Uncapacitated Facility) Location Problem (SPLP).
13. Cherenin-Khachaturov's Theorem. Excluding vs Preservation Rules. Dichotomy (Preliminary Preservation) Algorithm.
14. Branch-and-Bound (BnB) Algorihm for Supermodular Function Minimization Problem and Its Applications to the SPLP.
15. Non-binary branching rules applied to the pseudo-Boolean formulation of the SPLP. The branching rule: Make Quadratic Terms Linear.

16. The Quadratic Cost Partition Problem (QCP) - an example for maximization of a submodular set function.
17. Six equivalent definitions of submodular set functions.
18. Cherenin's Theorem (quasi-concavity of submodular functions).
19. The greedy algorithm for submodular set functions.
20. Boolean and pseudo-Boolean polynomials.
21. Truncation Theorem for the p-Median problem in pseudo-Boolean formulation.
22. Overview of BnB type Algorithms: Branch and Cut, Cutting Planes, Branch and Cut and Price, Data Correcting Algorithms, Tolerance Based Branch and Bound, Climer and Zhang's cut-and-solve algorithms.
23. The theory of NP-Completeness; Relationship between P and NP classes; Polynomial transformations from a language (problem) to another language (problem).
24. Cook, Karp, and Levin's Theorem (sketch of the proof). Algorithm for proving NP-Completeness.
25. Six basic NP-Complete problems: 3-Satisfiability, 3-Dimensional Matching, Vertex Cover, Clique, Hamiltonian cycle (circuit).
26. Overview of the Algorithms and Data Structures 2 course.

7. Homework will be based on assignments performed individually:

Homework problems will be assigned starting from first week just once a week. **Late submission will be accepted with penalty**. Students using the work of others without giving credit (plagiarism) will receive a failing grade in the course. Students will required to implement the learned algorithms and submit them to an automatic control system to pass the set of tests. Solution will be accepted only if all tests are passed.

8. Test and Exam

You will have one midterm test and a final exam (comprehensive). Midterm test will be announced at least one week in advance. A student who misses an exam (test) may make it up if prior permission is obtained from the Administration of Faculty of Computer Science.

9. Student's Presentation (at most 10 min = 8min + 2 min for questions):

The presentations will be based on team (individual) reports. Students will be credited for both their presentations and questions asked during the presentations.

10. Grading: ADS 2 Fall 2019-20

Homework and Assignments	20%
One test in-class	30%
Personal presentation of team reports	10%
Final exam (comprehensive)	40%

Here is an example to explain how the final grade will be computed for a specific student, Ms. Bright.

Homework assignments and reports (HA) 20%
One test in-class (OT) 30%
Personal presentation of team reports (PP) 10%
Final exam (comprehensive) (FE) 40%

Final Grade ADS 2 (FGADS2_NAME)= 0.4(FE)+0.1(PP)+0.3(OT)+0.2(HA),
for example, the following grades are received by the student Bright

HA = 10, OT=7, PP=9, FE=6, thus
FGADS2_Bright=0.4(6)+0.1(9)+0.3(7)+0.2(10)=
2.4+0.9+2.1+2=8.4 which will be rounding up to 9.

11. All grades will be posted online. Students should check their grades periodically and notify the instructor immediately if they notice any discrepancy.

Grade Scale

10	>=90%
9	>=80%
8	>=75%
7	>=70%
6	>=65%
5	>=60%
4	>=50%
3	>=40%NS)
2	>=25%
1	>=15%
NS	<50%

12. Mid-Semester Progress Reports will be issued for all undergraduate students halfway through the semester. You will be issued an S/Satisfactory – which means you are passing the class with a 4 or better; an NS/Not Satisfactory – which means your grade is a 3 or lower; or an FA/Failure due to absences. If you are in the NS or FA categories, you should contact your instructor and advisor to discuss ways to improve your performance before it is too late.

13. Literature

1. M.R. Garey, D.S. Johnson. Computers and Intractability. A guide to the Theory of NP-Completeness. W.H. Freeman and Company, San Francisco, 1979.
2. B. Goldengorin, P.M. Pardalos. Data Correcting Approaches in Combinatorial Optimization. Springer, NY, 2012.
3. Robert Sedgewick and Kevin Wayne. Algorithms. Pearson Education, Inc. 2011.
4. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms, Second Edition. The MIT Press, Cambridge , Massachusetts London, England McGraw-Hill Book Company, 2001.
5. B. Goldengorin. Lecture Notes 1, 2, 3.
6. G.L. Nemhauser, L.A. Wolsey, and M.L. Fisher. An analysis of approximations for maximizing submodular set functions 1. Mathematical Programming, 14 (1978), 265-294.
7. Sharlee Climer, Weixiong Zhang. Cut-and-solve: An iterative search strategy for combinatorial optimization problems. Artificial Intelligence 170 (2006), 714-738.
8. More References will be provided for seminars and personal student's presentations (talks).

The common mistakes made in the homework will be discussed during the seminars.
Homework and exams consist of tasks that are equivalent or similar to those which have been studied at lectures and seminars. Any fact of cheating or breach of academic integrity will result in receiving a "0" (zero) for this work.

Rounding of the cumulative grade, intermediate and final grades must be performed according to the following rules. Rounding down for marks between 1 and 5, rounding by the rules of arithmetic for marks between 5 and 6, and rounding up for all the other marks.
There is no possibility to get an extra point to compensate the low cumulative grade.

6. Methods of Instruction

The discipline is delivered through lectures seminars, including computer classes.