RUTGERS
Rutgers Business School
Newark and New Brunswick

Intro to Algorithms and data structure	**Dr. Alexander Tzanov**
Fall 2017. 26: 198: 685: 01 Index: 21661	Office Building
Room Number:	
Class Building & Room Number [& Campus]	Office Phone Number
Class Meeting Times: Fri:6:00 – 9:00PM	Email adress:alex.tzanov@nyu.edu

COURSE DESCRIPTION:

The algorithms are cornerstone of Computer Science and Information Technology (IT). To be successful every expert of researcher in IT should have current and working knowledge of algorithms. A particular problem can be solved in many different ways and with use of different algorithms, so is critical students to be able to analyze and compare different algorithms in order to choose the most effective algorithm for the problem. That requires deep knowledge of fundamental algorithms and essential data structures. The course study most useful fundamental algorithms and analyze their complexity. The objective is to gather insight into principles of algorithms' design and data structures useful in this design.

COURSE MATERIALS

The textbooks are:

1. **(Required)** T. Corman, C. Leiserson, R. Rivest, and C. Stein "Introduction to Algorithms", Third edition MIT Press 2009.
2. **(Optional)** Data Structures and Algorithms 1st Edition Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA ©1983
3. **(**Reference) R. Sedgewick, and K. Wayne *"Algorithms" Fourth edition,* Addison-Wesley, 2011

LEARNING GOALS AND OBJECTIVES

This course is designed to help students in business to develop skills and gather knowledge in analysis of algorithms. The students will acquire deeper understanding and will gather ideas for computational problems arising in practical applications – sorting, searching, divide and conquer, graphs and some others. By the end of the course the students who complete the course should be able to:

- Analyze given algorithm with respect to efficiency.
- To contrast and compare 2 algorithms with like function with respect of
    - Initial state of a data set the algorithm is suitable for
    - Best worse and average cases.

PREREQUESTS

Some programming experience.

ACADEMIC INTEGRITY

I do NOT *tolerate cheating*. Students are responsible for understanding the RU Academic Integrity Policy (https://slwordpress.rutgers.edu/academicintegrity/wp-content/uploads/sites/41/2014/11/AI_Policy_2013.pdf)

REQUIRED WORK

There will be 5 homework during the semester plus one extra credit homework. The total homework will constitute 36% of your final grade. Failing to provide 2 homework will result in fail. The midterm will worth 24% of your grade. There will be 1 project study and a final. The latter is worth 40%.

GRADING POLICY

Students are allowed to drop the extra credit homework. Fail on midterm or final will result in fail of the course.

TOPICS:

Topic 1. Introduction and basics – Read chapter 1,2,3 (Corman)

- Data structure
- Abstract data type
- Introduction to algorithms
- RAM model
- Asymptotic notations

Topic 2. Divide and conquer approach, recursion randomized algorithm – Read chapter 4,5

- Recursive relations and recursive algorithms
- Strassen algorithm for matrix multiplication
- Substitution method
- Analysis of algorithm performance through recurrence analysis
- Recursion tree method for solving recurrences.

Topic 3. Data Structures – Read chapter 10,11,12,13,18

- Stacks and queues
- Linked lists
- Binary trees
- Binary search trees,
- Balanced binary search trees,
- Hash tables and hash functions
- Red-black trees

Topic 4. Graphs algorithms. – Read Chapter 22,23,24,25

- directed and undirected graphs
- adjacency, and incidence matrix representation,
- depth- first and breadth-first search
- minimum spanning trees
- Prim's and Kruskal's algorithms
- Shortest path algorithms – Djikstra, Floyd-Warshall

Topic 5. Matrix operations and linear programming – Read chapter 28,29.

- Solving linear equations
- Inverting matrices
- Problems as liner programs
- Simplex method

Topic 6. Polynomials, strings and the FFT – Read chapter 30,32

- String matching algorithm
- Rabin-Karp algorithm
- Knutt-Moris-Pratt Algorithm
- Representing polynomials
- DFT and FFT.

Topic 7. NP-Completeness – Read chapter 34

- Turing machines
- Decision problems
- Optimization problems, and functional problems: Why focusing on decision algorithms is useful
- The first NP-complete problem: Satisfiablity problem (SAT), and refinements: 3-SAT
- NP-completeness of CLIQUE, TSP, COLORING, SUBSETSUM, KNAPSACK, INTEGER PROGRAMMING