# BOSTON UNIVERSITY METROPOLITAN COLLEGE COMPUTER SCIENCE DEPARTMENT

# MET CS 566 ANALYSIS OF ALGORITHMS

## Course Overview

Algorithm design and analysis provide the theoretical background for designing and analyzing algorithms including sorting, searching, dynamic programming, greedy algorithms, graph algorithms (shortest path, spanning trees, tree traversals), etc.

**Course Format** On Campus
**Time and location** Thursday 6:00 PM – 8:45 PM, Room CGS 525

## Prerequisites

MET CS 248 Discrete Mathematics and MET CS 341 or MET CS 342 Data Structures or instructor's consent

## Learning Objectives

This course attempts to provide an overview of the theory of algorithm design and analysis, and encourage students to develop and implement algorithms for various practical applications. Topics to be covered include Asymptotic notation; Sums and Recurrences; Divide and Conquer; Dynamic Programming; Greedy Algorithms; Graph Algorithms; Advanced Data Structures; Machine Learning algorithms and NP Completeness for decision problems.

Upon completion of this course, the student should learn:

- Principles of algorithm analysis and design
- Classification of functions by their asymptotic growth rate
- Abstract Data Type (ADT) specification and design techniques
- Elementary ATDs: lists, trees, stacks, queues, and dynamic sets
- Recursive procedures
- Induction proofs
- Principles of proving correctness of procedures
- Sorting algorithms (Insertion Sort; Divide and Conquer; Quicksort; Mergesort; Heapsort; and Radix Sorting)
- Dynamic sets and searching algorithms, including Hashing
- Graphs and Graph Traversals, including Depth-First Search algorithms on directed and undirected graphs

- Graph optimization problems, greedy and Minimum Spanning Tree algorithms; Huffman codes
- All-Pairs Shortest Paths in Graph algorithm
- Principles and techniques of dynamic programming, including the method of constructing optimal binary search trees
- Basic Machine Learning Algorithms
- Overview of NP-complete decision problems and approximation algorithms

## Required Textbook

Sara Baase, Allen Van Gelde, "Computer Algorithms", 3rd Ed.,
Addison-Wesley,2000, ISBN-0-201-61244-5

**Recommended book** T.H. Cormen, C.E. Leiserson,
R.L. Rivest, and C. Stein, "Introduction to Algorithms,"
3rd Ed., MIT Press, 2009, ISBN-13: 9780262033848.

## Courseware

Only a part of the lecture notes will be available in traditional digital-file formats (*.doc, *.ppt, and *.pdf) on the Blackboard site related to our course. Some handouts will be distributed in class and some notes will be written on the classroom board. Please, take notes every class and share the notes if someone needs them.

Blackboard Learn, URL: https://learn.bu.edu

## Evaluation and Grading

There will be two exams. If any grading criteria event will be missed it will be the responsibility of the student to arrange a mutually agreeable schedule for completion of work.

The grade is made up of your performance on your home works, project paper, midterm, final exams and participation.

Participation will be based on the percentage of in-class polling questions answered. Correctness of in-class polling responses will not be taken into account for participation grades.

Approximate weightings are as following:

| Assignments | Percentage |
|---|---|
| Six Written Homework Assignments | 48% |
| In Class Participation | 4% |
| Midterm Exam | 24% |
| Final Exam (or Project Paper) | 24% |

## Letter Grade:

| | | | |
|---|---|---|---|
| 94 ≤ G: | A, | 77 ≤ G < 80: | C+ |
| 90 ≤ G < 94: | A- | 73 ≤ G < 77: | C |
| 87 ≤ G < 90: | B+ | 70 ≤ G < 73: | C- |
| 83 ≤ G < 87: | B | 60<G<70: | D |
| 80 ≤ G < 83: | B- | G<70 : | F |

There will be no make-up exam for the final exam. If a student cannot take the final exam on the designated day, she/he will receive an incomplete grade.

## Academic Honesty

The course is governed by the Academic Conduct Committee policies regarding plagiarism (any attempt to represent the work of another person as one's own). This includes copying (even with modifications) of a program or segment of code. You can discuss general ideas with other people, but the work you submit must be your own. Collaboration is not permitted. See link below

http://www.bu.edu/met/metropolitan_college_people/student/resources/conduct/c ode.html.

## Instructor Information

Dr. Alexander Belyaev
Computer Science Department,
Metropolitan College Boston
University, 808 Commonwealth
Ave, Room 250 Boston, MA 02215
Office: 617-353-2566,
Email: abelyaev@bu.edu

Office Hours: After each class meeting

## Tentative Schedule

- The schedule is subject to change according to the actual progress of the class. Some topics may be skipped and some topics may be added.

- Students are encouraged to review book chapters assigned for each lecture before coming to class.

| SESSION | TOPIC | READING | HOMEWORKS |
|---|---|---|---|
| **1** (09/05) | Preliminaries (Chapters 1-2): What is an algorithm? Principles of algorithm analysis and design. Classification of functions by their asymptotic growth rate. Big-*O* and similar notations. Abstract Data Type (ADT) specification and design techniques. | Ch. 1, 2 | |
| **2 (09/12)** | Elementary ATDs: lists, trees, stacks, queues, and dynamic sets. Recurrence equations and their solution. Applications of recurrences. Induction proofs and examples. Principles of proving correctness of procedures. Review of series and sets, some probability. | Ch. 2, 3 | **Homework #1 due** |
| 3 (09/19) | Sorting Algorithms (Heapsort, quicksort, mergesort, and radixsort). Proof that sorting (by comparisons) takes *O*(n log(n)) time. Medians and order statistics. | Ch. 4 | |
| **4 (09/26)** | Divide and conquer techniques for mergesort and quicksort. | Ch. 4, 5 | **Homework #2 due** |
| **5** (10/03) | Dynamic Sets: Amortized time analysis; | Ch. 6 | |
| **6 (10/10)** | Graph algorithms: Spanning trees or forests; Shortest paths; and Maximum flow. Mid-term Exam Preparation | Ch. 7-9 | **Homework #3 due** |
| 7 **(10/17)** | **[MID-TERM EXAM] Ch. 1-7** | Ch. 1-7 | **MID-TERM EXAM** |
| 8 (10/**24**) | Graph optimization problems, greedy and Minimum Spanning Tree algorithms; Huffman codes | Ch. 8 | **Homework #4 due** |
| **9** (10/31) | Dynamic programming; Examples and principles**;** | Ch. 10 | |
| **10 (11/07)** | String matching: Rabin-Karp and Knuth-Morris-Pratt algorithms. | Ch. 11 | **Homework #5 due** |
| **11** (11/**14**) | NP-complete problems. Neural Networks | Ch. 13 | |
| **12 (11/28)** | Thanksgiving | | |
| **13(12/05)** | Deep Machine Learning; Virtual Reality. | Slide Presentation | **Homework #6 due** |
| **14(12/12)** | Miscellaneous topics and review. Project Presentations | Handouts | |
| **15 (12/16)** | **[FINAL EXAM] Ch. 8 - 13, or Project Paper** | Ch. 8 - 13 | **[FINAL EXAM] or Project Paper** |

## Communication

- All official announcements will be made in the class.
- All assignments will be posted on the class web page.
- **Important:** The primary method of communication is through in-class announcements. So, if you miss a class you need to talk to a friend in the class or contact me to find out

whether there was any important announcement.