Subject Code and Course Number: CS 211
Course Title: Introduction to Algorithms
Number of Credits: 3
Instructor Name: Armen Kostanyan, *Teaching Associate*: Nareh Salmasian
Email Address: armko@aua.am, TA:
Office Location: room 337
Office Hours: Saturdays, 10:00-11:30
Term/Year: Spring, 2018
Class Schedule: Tuesdays, Thursdays: 9:00 AM - 10:20AM
Prerequisites: CS 121 Data Structures
Co-Requisites: None

Course Description:

The course surveys topics including: review of main abstract data types; sorting algorithms, correctness, space and time complexity; hashing and hash tables, collision resolution strategies; graph algorithms; divide-and-conquer algorithms, dynamic programming; NP-completeness. Additionally we will consider some advanced data structures (such as RB trees, data structures for disjoints sets) and order statistic algorithms. Students are required to critically analyze, formulate and solve problems using analytical knowledge related to algorithms. Students should also be able to display proficiency in designing and analyzing complex algorithms and understand the software relevant to this field. Three hours of instructor-led class time per week including discussions and problem sets.

Required Materials:

- Textbook [CLRS]:
  - *Introduction to Algorithms*, *3rd edition*, by Thomas, H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, The MIT Press, 2009
- Recommended Literature:
  - R. Sedgewick. *Algorithms in C++. Parts 1-4: Fundamentals, Data Structures, Sorting, Searching*, Addison-Wesley, 1999
  - R. Sedgewick. *Algorithms in C++. Part 5: Graph Algorithms*, Addison-Wesley, 2002.

Schedule & Topics

Week	Topic	Reading	Non-Reading Home Tasks
1	Introduction ▪ Time and space complexity ▪ Asymptotic notation	CLRS: 1-3 [pp. 5-64]	hw1
2	Sorting and order statistics ▪ Quicksort ▪ Order statistics	CLRS: 7, 9 [pp. 170-190] [pp. 213-227]	hw2
3-4	Elementary data structures ▪ Binary heaps ▪ Binary search trees ✓ *Quiz 1*	CLRS: 6, 12 [pp.151-169] [pp.286-307]	hws 3-4
5-6.1	Red-black trees ▪ Definition, properties ▪ Operations ▪ Dynamic order statistics	CLRS: 13 [pp. 308-338]	hw5
6.2-7	Hashing ▪ Hashing with chaining ▪ Open addressing ✓ *Quiz 2*	CLRS: 11 [pp. 253-277]	hw6
	First examination		

8	Recurrences ▪ The substitution and recursion-tree methods ▪ Master method and its modifications	CLRS: 4 [pp. 83-113]	hw7
9	Divide and conquer algorithms ▪ The divide and conquer paradigm ▪ The maximum subarray problem ▪ Strassen's algorithm for matrix multiplication	CLRS: 4 [pp. 65-83]	hw8
10-11	Dynamic programming ▪ Rod cutting problem ▪ Matrix chain multiplication ▪ Finding a longest common subsequence ✓ *Quiz 3*	CLRS: 15 [pp. 359-397]	hws 9-10
12	Data structures for disjoint sets ▪ Linked list representation ▪ Forest representation	CLRS: 21 [pp. 561-572]	hw11
13-14	Graph algorithms ✓ Kruskal's and Prim's MST algorithms ✓ Dijkstra's SSSP algorithm ✓ Floyd-Warshall's APSP algorithm ✓ *Quiz 4*	CLRS: 23, 24, 25 [pp. 624-642] [pp. 658-664] [pp. 693-700]	hws 12-13
15	NP-completeness ▪ *P*- and NP- classes ▪ Sample NP-hard problems		
	Second examination		

Student Learning Outcomes:

The following chart shows alignment between course-specific and program learning outcomes and goals as identified in Program Curriculum Map.

General Education / University-wide Program Goals:

Course-based Student Learning Outcomes In this course, students will be able to:	***Program Student Learning Outcomes*** Students will be able to:	***Program Goal***
1.1.1. Assess the resources needed to solve the problem 1.2.1. Use classical algorithms and data structures to solve the problem; use algorithm designing techniques to develop own algorithms 1.3.1 Proving correctness of the algorithm; analyzing the efficiency of the algorithm in terms of its space and time complexity	1.1. Formulate and critically assess problems, including reviewing existing data and results on related topics as well as conducting a requirements analysis (Intermediate Level) 1.2. Design and implement solutions using development tools in keeping with state-of-the-art technologies (Intermediate Level) 1.3. Analyze the obtained solutions in terms of accuracy, performance and compliance with requirements (Intermediate Level)	*1. To provide the knowledge and skills to design, implement and manage the development of efficient computing systems that solve real-life industrial and academic problems*

4.1.1 Know how to refine the existing computing system for efficient data processing 4.2.1 Know applications of studied algorithms; know what algorithms can be used for efficient data processing	4.1. Understand the impact of computing systems in an scientific, economic, environmental and societal context (Beginning Level) 4.2. Supplement core CIS knowledge with increased awareness in a related field of study (Intermediate Level)	*4. To broaden the professional horizons beyond the conventional field's limits in search of cross-disciplinary areas of application*

Course Structure

Instructor-led class will meet twice per week (class attendance is mandatory). Weekly home works consist of a set of tasks related to studied material. Apart from the homeworks course includes 4 quizzes, midterm and final examinations.

Method of Evaluation

Student learning will be evaluated on the basis of the following weighted components:

- Class attendance and activity (10% of final grade)
  Students are expected to attend classes and demonstrate activity in the affairs relating to the course material. The rubric below describes three levels for the class attendance and activity.

  Class Attendance and Activity Rubric

	Excelent	**Good**	**Poor**
Class attendance and activity	Regular class attendance; raising thoughtful questions demonstrating deep knowledge of the course material; solving the majority of problems offered during classes	Almost regular class attendance; often asking questions; solving a considerable number of problems offered during classes	Irregular class attendance; occasionally asking questions; engaging in outside affairs during classes; ignoring the problems offered during classes

- Completeness and quality of homework assignments (20% of final grade)
  Format: problem solving/coding

  For assimilation the course material and preparation to quizzes/exams the almost weekly homework assignments will be suggested. Students are encouraged to attend office hours to analyze solutions of the previous tasks and discuss next assignments. It is strongly recommended that students work on the homework assignments on their own. If student closely collaborates with someone on solving of a problem, then he/she must provide the name of the collaborator at the submission sheet. Similarly, if student finds a solution to a problem either in the Web or in a book, he/she must provide this source. In any case, students must always write the solution in their own words. Failing to do so will be considered as a violation of assignment policy.

- Grades on quizzes (20% of final grade)
  Format: problem solving

  The course includes 4 quizzes covering the essential parts of course material. The quizzes are intended for assistance the midterm and final examinations.

- Grade on first examination (25% of final grade)
  Format: problem solving

  The first examination will cover the topics taught in weeks 1-7 and summarize the quizzes 1 and 2.

- Grade on second examination (25% of final grade)
  Format: problem solving
  The second examination will cover the topics taught in weeks 8-14 and summarize the quizzes 3 and 4.

Students who are caught cheating on either a quiz or exam (specifically, who provided incredibly identical solutions for a problem) will receive zero points in relevant parts.
The general principles for assessment of solutions of the problems included in either homework assignments, quizzes or exams are provided below.

Problem Solving Rubric

	Excellent	**Good**	**Poor**
Problem solving ability	Mastery of standard methods of solving problems; use of nonstandard approaches when needed	Knowing of standard methods to solve problems; using them properly	Insufficient knowing of problems solving techniques; often improper use them
Information using	Deep knowledge of the course material; possession of advanced knowledge	Possession of the course material	Insufficient knowledge of the course material
Problem answering	Solving problems correctly and providing exhaustive explanations	Solving problems properly	Solving of a small number of problems with poor explanations or without explanations at all
Coding	Providing a working code with a set of test cases	Providing a working code	Providing a compiled but partially working code

The final grades will be defined following to the ranges below:

Grade	**Grade Point**	**Percentile Range**
A+	**4**	**95 – 100**
A	**4**	**87.5 - 94.99**
A-	**3.7**	**82.5 - 87.49**
B+	**3.3**	**77.5 - 82.49**
B	**3**	**70 - 77.49**
B-	**2.7**	**65 - 69.99**
C+	**2.3**	**60 - 64.99**
C	**2**	**52.5 - 69.99**
C-	**1.7**	**47.5 - 52.49**

D+	1.3	42.5 - 47.49
D	1	35 - 41.49
D-	0.7	30 - 34.99
F	0	0 – 29.99

Library and Media/Technology Use

Students are encouraged to use supplemental online and reference materials available at the library to enhance their overall learning in the course. Students are encouraged to use audio-visual aids and presentation software as appropriate. If students have any questions or need additional support in using library resources or technology, they should confer with library staff, ICT, or the instructor.

Late Policy

Students are required to make homeworks and to take quizzes and exams when they are scheduled by the instructor. Late submitted homeworks will not qualify for grading; missed quizzes will not be reimbursed.

Make-up Procedures

In the event that a student missed an exam, the instructor is under no obligation to give a make-up. However, if the student brings a valid reason for absence, the instructor may decide to organize a make-up procedure. To take advantage of this opportunity, the instructor should be notified in advance about the emergency.

Policy on Grade Appeal

Students are entitled to appeal grades in line with the university's Grades Policies policy which is available online at http://policies.aua.am

Standards for Academic Integrity

Students are required to conduct themselves in an academically responsible and ethical manner in line with the Student Code of Ethics. Acts of academic dishonesty impair the academic integrity of AUA and create an unfair academic advantage for the student involved and other member(s) of the academic community. These acts are subject to disciplinary measures as prescribed in the AUA Code of Student Ethics http://policies.aua.am/policy/10
The Student Code of Conduct can be found at http://policies.aua.am/policy/101

Special Needs:

Students requiring special accommodations for learning should contact the Center for Student Success by the end of the Drop/Add period with such requests. studentsuccess@aua.am, http://studentsuccess.aua.am/disability-support-services/