Instructor: Antonio Miranda
Hill 363
Phone 57477
email: antonio.miranda@cs.rutgers.edu
Office hours: TBD

In this course we will study a variety of algorithms and analyze their complexity to gain insight into the principles and data-structures useful in algorithm design.

Textbook 1: *Algorithms* by Sanjoy Dasgupta, Christos Papadimitriou and Umesh Vazirani, McGraw-Hill.

Textbook 2: *Introduction to Algorithms* by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, 3rd Edition, MIT Press.

Students are responsible for knowing all the material that will be covered in class and is NOT in the book. Some lecture notes will be posted on Sakai.

Prerequisites:
Calculus and Discrete Mathematics.
Chapter 0 of Textbook 1, **Chapters 1, 2, and 3** of Textbook 2

Class Policy:

- **Exams.** The only acceptable reason for not attending an exam is a major (documented) medical emergency. NO make-ups will be given in any other case. A list of specific topics will be posted before each exam, to help you prepare.
- **Quizzes.** At the beginning of most lectures you will take a short quiz, usually it can be done in about 10 minutes.
- **Homeworks.**
  - Homeworks will be posted on Sakai.
  - Homework submissions MUST be handwritten.
  - All the pages must be scanned into ONE single file
  - The file must be uploaded to Sakai before the due date (time).
  - Make sure that your scanned any files are easily readable before submitting them.
  - NO late homeworks will be accepted

- **Regrading.** To report possible grading errors, attach a page describing the alleged error to the corresponding exam, homework, or project and submit it to the instructor or TA no later than one week after the date when the exam, homework or project was returned graded (to the class). An answer to a regrade may not be available until the end of the semester, so make copies of the materials given back for regrading. One week after the exam, homework or project was returned graded to the class, the grade becomes permanent and cannot be changed. The grade of the final exam becomes permanent one week after it is posted on Sakai.

- **Project.** The final project will be graded principally on functionality. In order to pass the course, a working programming project must be completed and handed in. Individual contributions to the project will be measured and taken into account, the instructor may request an oral examination to further evaluate a student's understanding of the material involved and the way in which the program works.

  The only communication between teams should concern very general topics such as how to log in, how to install software and the like. Reusing software written by others or for other courses/projects is prohibited, unless approved by the instructor.

- **Grading.**

Homeworks	15%
quizzes	5%
2 Midterm exams	30%
Final exam	30%
Final project	20%

  The grade assigned as final grade cannot be changed, even by doing additional work. In order to be fair to all students, any option to improve grades (if any) will be given to every student, NOT just to one particular student.

- **Academic Integrity.**

  We take academic integrity quite seriously. Copying answers from any source including published solutions is considered academic dishonesty.

  In case of learning disabilities, please provide verification from the College Coordinator. Also inform us at the beginning of the semester of any **planned absences** due to participation in professional events.

- **Sakai.**

  Sakai will be used for weekly announcements related to quizzes, homeworks, midterms, final exam, etc. If you are dully registered for the class you will get emails alerting you of these periodic sakai announcements.

Topics:

Topic	Description
Complexity Measures	Methods for expressing and comparing complexity of algorithms
Searching and Sorting	Lower bounds for comparison-based sorting; counting sort and radix sort
Divide and Conquer	Fast integer multiplication, recurrences, the Master Theorem, mergesort, median and selection algorithms, quicksort, heapsort, fast matrix multiplication
Graph Search Algorithms	Graph representations, depth first search, strongly connected components, breadth first search and layered DAGs
Shortest Paths in digraphs	Single-source shortest paths for non-negative edge weights, priority queues and Dijkstra's, single source shortest paths for general weights
Greedy Algorithms	Spanning trees and cuts, union-find and path compression, minimum spanning tree algorithms, sample randomized algorithms
Dynamic Programming	Shortest paths in DAGs, longest increasing subsequence, string matching (approximate), integer and (0,1) knapsack problems, chain matrix multiplication, all pair shortest paths, independent sets
Network Flows	Max-flow, bipartite matching
Introduction to Linear Programming	Duality, simplex algorithm
NP-Completeness and reductions	
Coping with NP-Completeness	Approximation algorithms, fixed parameter tractability
Algorithm Sampler (if time allows)	Some more advanced topics of current interest like Page Rank, External Memory Algorithms, Streaming Algorithms, Parallel Algorithms, Distributed Algorithms, and Quantum Computing

Important Dates:

Exam 1	2/25/2019
Exam 2	4/1/2019
Final Exam	TBD

HOMEWORK 1 due date: Wednesday 2/30/19 (in class)

Solve the following exercises from Textbook 1:

- 2.1
- 2.2

Quiz 1 Thursday 2/28/19

Those who want a SPN to enroll in this course must pass Quiz 1 with a grade of 60 or more