## Syllabus: CSCI 6212 Algorithms, Fall 2019

**Course Objectives:** This course presents the fundamental techniques for designing efficient computer algorithms, proving their correctness, and analyzing their running times. We will discuss a number of topics, including greedy algorithms, divide and conquer algorithms, dynamic programming, network flow algorithms, NP completeness and computational intractability, approximation algorithms, and randomized algorithms.

**Texts:**

Free Electronic textbook: http://jeffe.cs.illinois.edu/teaching/algorithms is a fantastic, free, online textbook from which there will be assigned readings.

Optional reference book: Algorithm Design, by Jon Kleinberg and Eva Tardos, Addison-Wesley, 2005.

Optional reference book: Introduction to Algorithms, (3rd Edition), by T. Cormen, C. Leiserson, R. Rivest, and C. Stein, McGraw Hill, 2009.

**Prerequisites** Each student is expected to have basic programming skills (programming with loops, pointers, structures, recursion), discrete mathematics (proof by induction, sets, permutations, combinations, and probability), understanding of basic data structures (lists, stacks, queues, trees, graphs, and heaps), knowledge of sorting algorithms (MergeSort, QuickSort, HeapSort) and basic graph algorithms (DFS, BFS, minimum spanning trees and shortest paths), and basic calculus (manipulation of logarithms, differentiation, integration), and exposure to previous algorithm design and analysis concepts. Please see me or the teaching assistant as soon as possible to head off any problems.

**Course Work** Course work will consist of (around 4) homework assignments (about one every week and a half) and 2 exams (a midterm and a comprehensive final). Tentative weights: Homeworks: 25%, Midterm: 30%, Final: 45%. (Note that these weights are subject to change.)

The Midterm will be held in class, but the exact date is yet to be determined (probably in mid to late October).

Homeworks must either be neatly written or typeset. I encourage you to use (and learn, if necessary), LaTeX, the default standard for technical writing. You can get a free account through "Overleaf", and the following tutorial can get you started:

`https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes#What_is_LaTeX.3F`

Poorly written work will not be graded. When writing algorithms be sure not only that your solution is correct, but also that it is easy for the grader to understand why your solution is correct. Part of your grade will be based not only on correctness, but also on the clarity, simplicity, and elegance of your solution.

Some homeworks will have a special challenge problem. Points from the challenge problems are extra credit. This means that I do not consider these points until after the final course cutoffs have been set. There is no official scale to extra credit points, but each semester extra credit points usually account for at least few students getting one higher letter grade (e.g., from a B+ to A-).

**Academic Dishonesty** All class work is to be done independently. It is best to try to solve problems on your own, since problem solving is an important component of the course, and exam problems are often based on modifications of homework problems. You are allowed to discuss class material, homework problems, and general solution strategies with your classmates. But, when it comes to formulating or writing solutions you must work alone. You may use free and publicly available sources, such as books, journal and conference publications, and web pages, as research material for your answers. (You will not lose points for using external sources.)

You must clearly and explicitly cite all outside sources and materials that you made use of. I consider the use of uncited external sources as portraying someone else's work as your own, and as such it is a violation of the University's policies on academic dishonesty. Instances will be dealt with harshly and typically result in a failing course grade.

Unless otherwise specified, you should assume that that the GWU Code of Academic Integrity applies.

**Topics:** The following list of topics is very tentative. Depending on time, some topics may be added or dropped, and the order of topics may change.

Review of algorithm analysis: Review of algorithm analysis (and summations and recurrences), review of basic graph theory and graph representations

Greedy Algorithms: Examples with Convex Hull, Greedy graph algorithms (Dijkstra and MSTs), Huffman coding

Divide and Conquer: Mergesort and inversion counting, closest pair

Dynamic Programming: Weighted interval scheduling, longest common subsequences, chain matrix multiplication, shortest paths in graphs (Bellman-Ford)

Network Flow: Network flows (Ford-Fulkerson), bipartite matching, edge-disjoint paths, circulations, applications

NP and Computational Intractability: Polynomial-time reductions, the definition of NP, NP-complete problems

Approximation Algorithms: Greedy algorithms and bounds on the optimum. Examples of approximation algorithms (vertex cover, travelling salesman, set cover)