CIS 5515 Design and Analysis of Algorithms

- **Catalog Description:**

  – Prerequisite: CIS 5511 (Programming Techniques): minimum grade of C; may not be taken concurrently.

  – The course objective is to provide students with an understanding of the principles and techniques used in the design and analysis of efficient algorithms. Emphasis will be on critical thinking, problem-solving, and rigorous analysis. The main topics cover Greedy Algorithms, Divide and Conquer, Dynamic Programming, Network Flow, Approximation Algorithms, Adversary Arguments for Lower Bounds, and Theoretical results related to NP-completeness. A variety of classic algorithms will be chosen for discussion throughout the course, as they are important and/or are helpful for understanding the fundamental concepts.

- **Textbook:**

  – *Algorithm Design*

  Jon Kleinberg and Eva Tardos, Addison-Wesley, 2005.

  References:

  – *Introduction to Algorithms*, Third Edition

  Thomas Cormen, Charles Leiserson, Ronald Rivest, and Clifford Stein, MIT Press, 2009.

- **Instructors:**

  – Dr. Jie Wu, Laura H. Carnell Professor, CIS, Temple University

  jiewu@temple.edu, www.cis.temple.edu/~wu

- **Goals:**

  – The student will get exposed to some advance problem solving skills, including design and analysis of algorithms, and research trends in this area.

- **Class time:**

  – Wednesday: 5:30 pm - 8:00 pm, Tuttleman Learning Center 0401A

- **Office hours**

  – Wednesday: 3:00 pm - 5:00 pm, SERC 362

- **Grading Policy:**

  – Homework: 6 * 5% = 30%

  – Midterm: 35%

  – Final: 35%

- **Prerequisite by topic:**

1. Basics of data structure and classic algorithms
2. Knowledge of a high level programming language
3. Elementary discrete mathematics

- **Tentative class schedule:**

1. Jan. 16 Introduction and algorithm analysis (Chaps. 1 & 2)
2. Jan. 23 Greedy algorithms I (Chap. 4)
3. Jan. 30 Greedy algorithms II (Chap. 4)
4. Feb. 6 Divide and conquer I (Chap. 5)
5. Feb. 13 Divide and conquer II (Chap. 5)
6. Feb. 20 Dynamic programming I (Chap. 6)
7. Feb. 27 Dynamic programming II (Chap.6)
8. March 2 - 8 (Spring break)
9. March 13 **Midterm Examination**
10. March 20 Network flow I (Chap. 7)
11. March 26 Network flow II (Chap. 7)
12. April 3 NP and computational intractability I (Chap. 8)
13. April 10 NP and computational intractability II (Chap. 8)
14. April 17 Approximation algorithms I (Chap. 11)
15. April 24 Approximation algorithms II and adversary augments (Handouts)
16. April 30 - May 2 (Study days)
17. May 8 **Final Examination**