Advanced Algorithms*

COT 5405 (Section 1)†
Piyush Kumar

Handout #1, – Course Information

Course Web Site. `http://www.compgeom.com/~piyush/teach/AA05/`

Class Mailing List. Announcements for the course, homeworks, reading assignments, programming projects will be available using the blackboard (`http://campus.fsu.edu`). Make sure you check both the course web site and the blackboard at least once in two-three days throughout the semester.

Instructor. Piyush Kumar.
URL: `http://www.compgeom.com/~piyush`.
Office Hours: Wednesday, 6:30 to 7:30pm. Alternatively, you may schedule an appointment, either by email or by phone.
Phone: 645-2355
Email: `piyush@acm.org`
Venue: Office Hours will be held at Love 105B (My Office)

Lectures. Monday, Wednesday at 5:15pm to 6:30pm at Love 0103.

Exams. Finals exams will be held on Dec 14th, 5:30 to 7:30pm.

Course rationale: This is an advanced course in design and analysis of algorithms covering topics typically not covered in undergraduate algorithms.

Course Description. Algorithms is an integral part of computer science and mathematics. So far, you have acquired proficiency in basic data structures, algorithms and programming. This course is the next step towards becoming an algorithm designer for the real world. We plan to cover the following topics in this course (tentative).

- Network Flows:
  - Matching and its applications (Programming Assignment). We will talk about stable machings and their applications. Applications we will talk about include Image segmentation, and disjoint paths in graphs. Our first programming assignment will be related to Matching.
  - Max-Flow Min-Cut Theorem. It's history. Flows using LP. Max-Flow Min-Cut using Ford-Fulkerson. Scaling Max-Flow algorithm. How to use Max-Flow to do maximum Bipartite matching.
  - Hall's Theorem, Menger's theorem.
  - Weighted Bipartite Matching and implmentation assignment in Boost.
- Complexity classes and Approximation Algorithms.
  - Cook/Karp Reductions, examples.
  - Define P, NP, co-NP, PSPACE, EXP.
  - Revision of intractable problem classes.

*Preliminary version. I will distribute the final version in the first class.
†Course Reference Number 09448

  - Vertex cover, Independent Set, Hitting Set, k-coloring, Set Cover, Set Packing, Hamiltonian Cycle, 3-SAT, Circuit SAT.
  - Approximation Algorithms. Examples like TSP, Vertex Cover, k-center.
  - Hardness of Approximation for TSP.
- Optimization:
  - Linear Programming: Geometric Intuition, Duality, Simplex, Approximation algorithms using LP, Weak Duality, Complementary Slackness, Primal-Dual Algorithms, Integer Programming, LP Relaxations, Vertex Cover using LP.
  - Clustering problems: Low Diameter Clustering, Hierarchical Clustering, k-center clustering, clustering using MSTs.
- Computational Geometry:
  - Range searching (Range Trees), Fractional Cascading, Convex hulls (Graham Scan, Jarvis March), Randomized incremental constructions, Low Dimensional LPs, Intersection of convex polygons in linear time.
  - Introduction to output-sensitive algorithms
  - Finding a large triangle inside a polygon (Programming Assignment).
- Data Compression: Entropy, Lossless/Lossy Compression, RLE, Entropy, Kraft-McMillan Inequality, Huffman Coding, Proof of optimality, Arithmetic coding, LZ77, Residual Coding, Move-to-front Coding, RLE, JPEG-LS, Facsmile ITU T4, Lossy coding, Scalar/Vector Quantization, Transform Coding.
- D&C:
  - Counting Inversions
  - Closest Pair of points
  - Integer Multiplication: Karatsuba Multiplication
  - Matrix Multiplication, Cache Aware MM, Cache Oblivious MM, Cache Oblivious Matrix Transpose, Van Emde Boas Layout for static searches, Introduction to External memory algorithms using sorting.
  - DFT
- Introduction to streaming algorithms.
- Advanced Data Structures:
  - Hashing.
  - String Matching: Tries, Compressed Tries.
  - Introduction to search engine design.
- Online algorithms: Online Dating, Ski Rental, Online Caching and Competitive Analysis.
- Parallel algorithms: Introduction to PRAM, Prefix Computation, Array Packing, MergeSort, Closest Pair, Convex hulls.
- Machine Learning: Perceptrons, Convergence proof, Connection to LP.

Learning Objectives. The objective of this course is to encourage you to learn how to :

- design and implement 'new' algorithms in the real world.
- map problems to algorithmic problems.
- read and understand algorithms published in journals.
- develop writing skills to present your own algorithms
- collaborate and work together with other people to design new algorithms.

Prerequisites. A Grade of B or better in COP 4531 or CGS 5427 or an equivalent course. Come and talk to me if you do not have the prerequisite and you still want to take the course. You will find basic concepts of combinatorics (counting, graphs, recursion) to be very useful. I will assume that you have taken an undergraduate algorithms course and need no introduction to quicksort, mergesort, $O()$ notation, shortest paths, Minimum spanning trees, basic data structures like arrays, trees, heaps etc. Finally, it is useful to have experience with C, C++, or Python. (You should be able to code in C++.) Some of the homeworks will ask you to write code. There will also be a programming project in the course.

Textbooks. I will assume that each of you own a copy of *Algorithm Design* (Hardcover) by Jon Kleinberg, Eva Tardos ISBN: 0321295358. You should also have access to [CLRS] book that I use in 4531. Most of the material in the course will be from the following books.

1. *T. Cormen, C. Leiserson, R. Rivest, and C. Stein.* Introduction to Algorithms. (2nd edition). MIT Press , McGraw-Hill, 2001.
2. *R. Motwani and P. Raghavan.* Randomized Algorithms. Cambridge University Press, 1995.
3. *V. V. Vazirani* Approximation Algorithms. Springer. 2001.
4. *Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin.* Network Flows: Theory, Algorithms, and Applications. Prentice Hall, 1993.

I have requested the above material to be put on reserve in the library. The text book is sold out at the FSU bookstore but is available at the Bill's on Copeland and at Bill's on Tennessee.

Course Policies

1. **Homeworks:** The best way to learn the material is by solving problems. You are encouraged to work in *pairs*[1], because the best way to understand the subtleties of the homework problems is to argue about the answers. If you do not have a partner, let me know and I'll try to hook you with one. If you want a divorce, you should let me know too. Don't be a leech and let your partner do all the work. Unless you learn how to solve problems, I *promise* that you will get burned on the exams and thus for your final grade.
2. Your solutions should be handed in *Stapled together*.
3. Your solutions should be *very neatly written*. If your solution is unclear, sloppy, or if your solution is hard to understand, you will have points deducted even if your solution is correct. One of the best way to make your solutions clear is to *include pictures* and *examples*.
4. Homework assignments will be due at the *beginning of class* and I will hand out the solutions immediately. **Late assignments** *will not be accepted* because the solutions will be available.
5. It is extremely important that you *start homework assignments early*. The homeworks are very challenging, and if you get behind in your work, you may find it too difficult to catch up. Out of all the graded homework sets, I will drop the min score before calculating the total homework score towards the final grade. Since I drop the lowest score, missing one homework due to an illness should not be a problem.
6. **Grading Criteria:** The grade for COT 5405 will be assigned based on the following approximate percentages.

	Approximate Percentage	Variable
Homework	10%	
Class Participation	10%	
Programming Assignments	10%	$h \in [0, 30]$
Midterm	20%	$e \in [0, 45]$
Final	25%	
Final Project	20%	$f \in [0, 25]$
Project Presentation	5%	

To Pass: $h \geq 16$ and $e \geq 22$ and $f \geq 13$.

Final Grades: Your final grades (letter grades) will depend on your $(h + e + f) \in [0, 100]$ score. There is no pre-established scale or curve. I will sort all the $(h+e+f)$ scores for those who pass and assign letter grades to different non-overlapping intervals (The highest level being A and decreasing thereof). I will at my discretion, use clustering to generate the intervals or the following intervals (Whichever yields you a *better* grade).

Percent	Letter	Percent	Letter	Percent	Letter	Percent	Letter
94-100	A+	84-87	B+	74-77	C+	64-67	D+
90-93	A	80-83	B	70-73	C	60-63	D
88-89	A-	78-79	B-	68-69	C-	57-59	D-
0-56	F						

[1] Students who have taken any of my previous courses are forbidden to be in the same pair.

7. I reserve the right to modify these numbers uniformly by 5% each. I reserve the right to de-emphasize the homework grades if there is evidence of students who copy instead of doing the homework themselves.
8. Scribing will be worth *approximately* 4 or 5 percentage points of extra credit (This option is only for people who know LaTeX and XFig or are willing to put the effort to learn it).
9. **Missed exam Policy:** A missed exam will be recorded as a grade of zero. We will follow the university rules regarding missed final exams (see `http://registrar.fsu.edu/dir_class/fall/exam_schedule.htm`), for all the exams, including the final exam.
10. **Grade of 'I' Policy:** The grade of 'I' will be assigned only under the following exceptional circumstances:
    - The final exam is missed with an accepted excuse for the absence. In this case, the final exam must be made up during the first two weeks of the following semester.
    - Due to an extended illness or other extraordinary circumstance, with appropriate documentation, the student is unable to participate in class for an extended period. In this case, arrangements must be made to make up the missed portion of the course prior to the end of the next semester.
11. **Academic Honor Code:** Because a primary goal of the course is to teach professionalism, any academic dishonesty will be viewed as evidence that this goal has not been achieved, and will be grounded for receiving a grade of F (You must read the FSU Academic Honor Code in the Student Handbook and abide by it). Copying/Modifying other people's programs/code will be treated the same as copying in an exam.
    - Every student must write his/her own code and homework. Showing your code or homework to members of other teams, giving it to them, or making it accessible to them (e.g., by making the files world-readable) is academic dishonesty.
    - You are responsible for ensuring that your code/documentation/results/homeworks are adequately protected and not accessible to others. Change permissions of your working directory to 0700 (chmod 0700 {directory}).
    - Consulting code from a textbook, or from the Internet, in order to understand specific aspects of your assignment is fine. However, *copying entire code or large parts of such code will be considered academic dishonesty*. If you borrow small parts of code from these sources, you must acknowledge this in your submission and additionally you must clearly understand and be able to explain how the code works.

    Once again: There is no excuse for cheating in any circumstances. See me before you even *contemplate* cheating.
12. **Accommodation for Disabilities:** If you have a physical, psychological, medical or learning disability that may impact on your ability to carry out assigned course work, I would urge that you contact the staff in the Student Disability Center and bring a letter to the instructor indicating the need for accommodation. The Student Disability Resource Center will review your concerns and determine, with you, what accommodations are necessary and appropriate. All information and documentation of disability is confidential. They can be contacted at `(850) 644-9566`.
13. **Attendance Policy:** The university requires attendance in all classes, and it is also important to your learning. The attendance record may be provided to deans who request it. If your grade is just a little below the cutoff for a higher grade, your attendance will be one of the factors that we consider, in deciding whether to "bump" you up to the higher grade. Missing three or fewer lectures will be considered good attendance. In rare cases, such as medical needs or jury duty, absences may be excused with appropriate documentation. You should let me know in advance, when possible, and submit the documentation I seek. You should make up for any materials missed due to absences.
14. **Syllabus Change Policy:** The syllabus is guide to the course and subject to change with advanced notice.

Final Project

1. You have to write a short (10 page) paper on your project. I will make the list of related projects for you to pick shortly after the beginning of the semester.
2. The paper will be due Nov 26th, 2005. You will also need to give a demonstration of your project by this date. There will be a short project presentation which I will schedule where you would explain what you did for the project and show your results.
3. The main idea behind the research project is to be creative either in designing a new algorithm, in implementing a research paper in a novel way, or developing an extention of the algorithms treated in the class. You should also make sure that you know about the relevant literature and cite it in your paper.
4. Preferably the paper should be typeset in LaTeX.
5. Try to select a project which interests you or with which you have previous experience (or both). Once you choose your project, make a plan of how to attack the chosen problem. Make a tentative project schedule with milestones and email me your milestones. 7% of your final project grade will depend on your project plan and if you abide by your plan.