

CIS 502: Algorithms

Course Information: Fall 2008

General Information

See webpage “www.seas.upenn.edu/~cis502” for the most recent information about office hours etc.

Exam/Homework Policies

1. Midterm is open “any handwritten material”. You cannot use any electronic devices.
2. **Final is closed book, closed notes.**
3. Give yourself time. What you take away from the course is how to think about problems, so try to reserve time for the homeworks.
4. You are strongly encouraged to try and solve the exercises at the end of the chapters.
5. *Extensions to homeworks and makeups for exams will only be given for verifiable medical reasons.*
 - We will have onehalf-weekly homeworks (due in about 10 days or so).
 - You are allowed to discuss the homework problems in pairs but you have to write your own solution. Everyone has to write the names of the person they worked with. **You cannot work with the same person for more than 1 homework.**
 - If there is an academic/Penn conflict, e.g., you are representing Penn and will be unavailable, you are presenting paper in a peer reviewed conference, email sudipto@cis as soon as you are aware of the conflict and definitely 3 days before the due date. Realize you can inform of such at the beginning of the term.
 - If you are interviewing/have personal reasons/have a difficult schedule, you are asked to plan the homework. Such request will likely be denied.
 - Homework is due at the **end** of the class on the due date, it can be submitted in class or at Levine 502 with Cheryl Hickey. There will be a 25% reduction in grades if it is turned in late, by 5pm the next working day. Any submission later than this time will not count towards your grade.
 - **You cannot use internet or newsgroup resources to solve the problems. Be mindful of asking clarifications in the newsgroup that cloak a search for solution in the newsgroup.** If in doubt, you can ask and the staff will answer as appropriate. If you use a book other than the text you **must** cite it – you will not be penalized in grade for the citation. Remember, if you can google it – so can we.

The following note will not concern most students. However, homeworks of two or more groups that appear to be very similar will mean that each of them will be asked to explain the situation as well as the homework solution on the blackboard in my office. If the explanations are not satisfactory, the involved persons stand to lose the **entire** homework grade which is 20% of the whole course and be forwarded to office of student conduct as well as other faculty in CIS.

Textbook and Syllabus

The textbook for the class is *Algorithm Design* by J. Kleinberg and E. Tardos, Addison Wesley.

- Chapters 1 and 2: Review material. We will cover this cursorily, you are expected to know this.
- Chapters 3–8: We will cover almost all of the material. The starred topics will be covered depending on the progress of the class.
- Chapter 13: Randomized algorithms

- Linear programming
- Chapters 11: We will try to cover as much as we can.
- Chapter 9, we will touch upon in connection to Chapter 8.
- Special topics: Computational Geometry (range searching, convex hulls, Voronoi diagrams).

Other Books: The following books may be handy if you feel you need more material beyond the class and the text.

- *Introduction to algorithms, 2nd ed.*, Cormen, Leiserson, Rivest, Stein. The first 14 chapters are a good review of what we expect you to know from undergrad classes.
- *The art of computer programming*, Knuth. A timeless classic.
- *Introduction to Algorithms: A Creative Approach*, Manber. I liked it as an undergrad; its an useful book to learn about the thinking process but will not have all the material.
- *Special topics: Randomized Algorithms*, Motwani and Raghavan; *Probability and Computing*, Mitzenmacher and Upfal; *Approximation Algorithms* Vazirani. *Network Flows*, Ahuja, Magnanti, and Orlin; and *Computational Geometry*, deBerg, Van Kreveld, Overmars, and Schwarzkopf.

Interesting Articles about theory and CS: *Selected Papers on Computer Science*, Knuth. A bit old, but a fantastic starting point for the algorithmic viewpoint.

Stylistic issues

An algorithm without a proof is a romantic novel. Exciting as it is, unfortunately it will get no points.

Prove/show/argue/give an algorithm = PROVE. What is a proof ? A proof is an argument that covers **all** cases. Often the difficulty of a problem is in one particular case; not handling all cases is an incorrect answer. You can say “this case is similar to...” **if** no other detail is necessary for that case. Be careful in using this.

A wrong answer is a wrong answer. Partial credit will be given but it is a subjective decision, you are encouraged to get the solutions right. Specially be careful with simple questions since it is almost impossible to decide if the answer was an omission or a typo – and there will usually be little partial credit.

Being a misunderstood genius is hardly fun. A significant part of any scientific effort is the ability of others to understand and reproduce them. All written work must be neat, well-organized, and include sufficient explanations. Messy, poorly organized, or illegible material will be returned ungraded.

A correct answer is typically short and clean. Providing a long winded discussion of all material relevant to the posed question (e.g., history of modern science), in the hope that the correct answer will be spotted, is not a useful strategy since that depends on the grader’s ability to spot answers. In such a case the credit will be given to the grader and not to the answer. Such answers usually demonstrate lack of clarity and understanding. If your argument **has** to be long winded; then provide a roadmap/intuition upfront. In a proof this corresponds to the explanation of the structure of lemmas you prove towards your theorem. **Often, this is as important as the proof itself.**