Computer Science CSC2414 Fall 1998
Prof. Daniel Panario University of Toronto

Topics for Course Projects

Due: Friday, 18 December 1998
Mark: 55% (minor project 5%; major project 40%; oral presentation 10%).

The following is a list of some possible topics for the course project. You can choose topics outside this list; in this case, you must talk with me and we should agree on the project. I strongly suggest you start your search for a topic that fits your interests very soon.

The kind of topics in the list vary from purely theoretical, to experimental. Your project may involve both components: some implementations together with some theoretical explanations. However, for some topics a survey with no implementations is more appropriate. Projects with implementations must include a report explaining how the constants were derived, and must contain well-justified data testing with a comparison with the expected mathematical behavior.

A portion of the marks go to how your project is written and organized. I suggest you consider the following scheme: include a title with an abstract. In a first section, explain the problem you are addressing, the background (if needed), and clearly state your results. No proof of theorems or programming code must appear in the first section. Then, describe the problem, the method you used, how and why it works, tables summarizing your experiments (if applicable) with clear explanations of the results. Finally, a list of references should appear. Programs (if applicable) should appear in an appendix.

We include a list of possible references for most of the topics. In general, this is intended as a starting point for the search but in some cases is self-contained. You can have copies of the papers or more precise references with me.

The project must be your own work. In particular, you must cite everything you are taking from the literature. You can take proofs, explanations, etc, from papers but the final writing must be only yours. One possible way to enhance (and show) your understanding of some work is giving new proofs of results, filling some missing steps in theorems, adding examples, and so on.

The due date is **December 18, 1998**. This is not a firm deadline and indeed, it is negotiable. In case you prefer to hand in your work in January, the firm deadline is **January 15, 1999**. You must have a meeting with the instructor to discuss your project before **November 5, 1998**.

Down to earth projects:

1. Survey of some asymptotical method not covered in lectures (Tauberian, Darboux, etc): main ideas, proofs, examples. Odlyzko; Knuth's books; Flajolet & Sedgewick.

2. Automatic average-case analysis of algorithms: implementation issues and potentialities. Flajolet, Salvy & Zimmermann.

Intermmediate level projects (understanding and clear explanation of the papers is enough; simple variations giving new results may be possible):

1. A comparative analysis (theoretical - experimental) a la Knuth (the algorithm must not be in TAOCP). Example: median of 2t+1 for Quicksort.

2. Random allocation problems (least-recently-used, move-to-front, etc) [for database minds]: Gardy et al.

3. Shellsort with several increments (Janson & Knuth; Yao).

4. Variants like Quickfind with median of 2t+1 and stopping with sizes smaller than M (Martinez, Prodinger, et al).

5. "The birth of the giant component": a detailed explanation [for graph theory people].

6. Variants on tree algorithms (like studies of ascendants and descendants, etc).

The following list includes some topics that may lead to future research. In general, these topics involved more elaborated concepts. If you decide to work on any of these problems, you must discuss it with the instructor to establish a reasonable cut off for the project. After the course is completed, you may (or not) pursuit this line of research.

1. Average-case analysis for polynomial problems over finite fields (gcds, etc).
2. General properties of random decomposable structures (sizes of components, etc).

and many more....