



Project kick-off

Algorithm Design II

Rasmus Pagh

Project - formalities

From course description:

"In the project, each student will consider 2-3 algorithmic problems (typically ones that can be seen as a variant or extension of a known algorithmic problem) and document their work on those problems in a project report."

Hand-in of report:

- No later than 3 PM on December 12.
- Physically in three copies at the exam office.
- Probably 15-20 pages + appendices with code.

Oral exam:

- January 10 and 11. No student presentation or prepared question.
- Instead, conversation with project report as starting point, which can touch upon other things treated in the course.

Project - setup

Groups:

- Ideal group size is **two**. Deviation of -1/+1 negotiable.
- Self-formed based on interests, previous group work experience,...
- Idea is to finalize group formation today.

Hand-in of report:

- Before 3 PM, December 12 (4 weeks, 4 hours, 44 minutes from now).
- Physically in three copies at the exam office.

Oral exam:

- January 10 and 11, individual exam of 20 min/student.
- No student presentation or prepared question.
- Instead, conversation with project report as starting point, which can touch upon other things treated in the course.

Problem formulation

Algorithmic problems:

- Each group should formulate 2-3 algorithmic problems that it wants to work on.
 - May or may not be connected.
- The problems should be "fresh", i.e., not identical to ones you have looked at before.
 - But OK to use one of the problems you have worked on so far as a starting point.
- The problems should be "tractable", i.e., within reach with the time and knowledge you have (Morten and I will help assessing this).
- Initially, your problems need not be precisely specified - this is part of the process that you will document.

Data set:

- Think about what data set(s) you could test your algorithm on.
- The course home pages has a number of links to possible data sets.

Project should be aligned with course goals

After the course, the student should be able to solve a wider range of real-life programming problems in a scalable way, by **employing algorithmic design techniques and tools**.

In particular, you should be able to:

- Identify and **formulate precisely** (if possible) the algorithmic problem hidden in a given programming task.
- **Theoretically analyze** the performance of a given algorithmic solution, including the analysis of basic approximation algorithms and basic randomized algorithms.
- At a basic level, evaluate theoretically the performance of an algorithm in a parallel or distributed setting, and in situations where there is a massive amount of data.
- **Find results in the algorithms research literature** relevant to a given problem.

What about the topics of coming lectures?

Consider letting one of your problems be related to things we did not yet cover:

More randomized algorithms:

- Hashing, load balancing
- Algorithms for finding “nearly identical” objects

Streaming algorithms:

Algorithms that just need to look at each data item once to:

- Estimate the number of distinct things seen,
- Find the most frequent data items in the stream,

Using very little space.

Massive data algorithms:

Algorithms tailored for working well in a memory hierarchy, e.g., when most data resides on a (slow) block storage device.

Problem formulation process

Timeline:

- Today:
 - Agree on what kind of problems and data sets you want to work on.
- Coming week:
 - Specify problem and work plan for project more precisely.
 - Find literature (books/papers) related to your problems.

Use of Piazza:

- We will use Piazza to work on the problem formulation process.
 - Each group posts a draft description that can be iteratively refined.
 - Morten and I will post feedback/questions that you can incorporate in a revision.
- Idea of open process is that you can be inspired by each other - but remember that each problem considered must be unique.

Group formation

Presentation round: Each of you says:

- Whether you already formed a group,
- What kind of thing you would like to work on.

Following the presentation round, people without a group should mingle and try to find someone with related interests.

Once groups are formed, you simply get to work!