Efficient Route Planning

SS 2012

Lecture 12, Wednesday July 25th, 2012
(Course evaluation, Exam, Work at our Chair)

Prof. Dr. Hannah Bast
Chair of Algorithms and Data Structures
Department of Computer Science
University of Freiburg

Overview of this lecture

- Organizational
  - Results and feedback for Ex. Sheet #11 (Transfer Patterns)
  - This is the **last** lecture … course evaluation summary
- Exam
  - Kind of exam questions to expect
  - We have a look at the exam from last year
- Work in my group
  - Working style
  - Current route planning projects
  - Other projects, in particular: search

Exercise Sheet #8 (Transfer Patterns)

- Summary / excerpts last checked July 25, 15:50
  - Basic idea of transfer patterns was clear
  - But not easy to understand all the details
  - In particular: why the need for the arrival loop?
  - File for Hawaii was not sorted
- Results of experiments
  - Three results on the Wiki, many were busy with other stuff
  - In those results ≈80% of station pairs had 1 – 20 TPs
    - and ≈50% of stations pairs had 10 – 20 TPs
  - Connectivity on Hawaii was fine, it seems

Course evaluation results 1/9

- First some statistics ...
  - Study program

    14 x Master, 3 x ACS, 2 x Bachelor (1 Info, 1 Math)
  - Size of audience
    - around 20 people still active towards the end
    - of those 19 submitted an evaluation form
    - Teaching award recommend.: 14, that is ≈75% **THANKS!**
  - All the details of the evaluation results on the **Wiki**
    - Next slides: summary of the main points, and ...
    - What was better than last year + further improvements

Course evaluation results 2/9

- Contents of the course
  - Very interesting and practical topics (many)

    "Especially the Google Maps API part was fun" (several)

    "I'm usually skeptical about algorithms lectures … often no visible usage for them … no way to apply in real problems … then you tend to stop caring about it. Here it was perfect."
  - Mix of theory and practice is just right (many)
  - Statistics for "I learned a lot in this course"

    15 x strongly agree, 3 x agree
  - Statistics for "The level of the lecture contents is ..."

    7 x rather high, 10 x appropriate

Course evaluation results 3/9

- Style of the course
  - Nice and relaxed atmosphere (many)
  - Very good and intuitive explanations (many)
  - Statistics for "The lecturer explains well"

    15 x strongly agree, 3 x agree
  - Good motivation to learn / do the exercises (many)

    "The course is perfect, a lot of work but a good way of getting experience in programming again and very motivating."
  - Examples / drawings were particularly helpful (many)
    - please no prepared drawings, better **live** (one)

Course evaluation results 4/9

- Exercise Sheets
  - Implementations really helped to understand it all (many)
  - Learned a lot about coding in general (several)

    "The course is about route planning, but really teaches so much more: good code design, efficient C++ programming, benchmarking and the fact that you always have to click twice on WebSVN links"
  - Exercises building on top of each other was a problem when you missed / had problems with one exercise (several)
  - Statistics for "The exercises are ..."

    14 x rather difficult, 4 x appropriate
  - Statistics for "How many hours a week for this course"

    12 x 9-12 hours, 5 x more, 1 x less

Course evaluation results 5/9

- Materials / Online offer
  - Video recordings are great (many)

    "High quality video recordings + prompt upload made it hard to find a real incentive to be present during the lecture"
  - Statistics for "How did you consume the course"

    8 x present, 4 x recordings, 6 x partly / both
  - Great and prompt support on forum (several)
  - Great course system: Daphne, SVN, ... (several)
  - Don't update slides shortly before the lecture (twice)

Course evaluation results 6/9

- Tutors
  - Comments were helpful and **much** appreciated (many)

    "The feedback on the exercises is outstanding. The tutors actually read and comment on the code in detail! Usually you get: "of course we don't read your code …"

    "My tutor did an amazing work correcting my exercise sheets. I got great advice from him. Also helped me to hunt complex hidden bugs."
  - Statistics for "The tutor explains well"

    11 x strongly agree, 5 x agree, 2 x undecided

Course evaluation results   7/9

- Miscellaneous
  - "You learn how to code well with the style checkers"
    - but should not care about spaces and such things   (one)
  - "The code becomes really ugly after a while"
  - "A lot of time went by with restructuring the code."
  - "Some topics in the lecture were not formal enough for me, so implementation was sometimes really hard."
  - "It would be nice if test cases would be provided."
  - "I'd prefer some more elaborate coding techniques to make more beautiful code, but I agree there's not much time."

Course evaluation results 8/9

- Improvements over last year's lecture
  - Overall, the lecture improved **a lot**
  - **Much** better structure + logical progression of material
  - For the complex algorithms: better split over two lectures
  - More formal + more precise + more complete
  - **Much** more (and more concrete) implementation advice
  - Eliminate boring parts, i.p. overly long coding or drawing
  - Simplified tasks for various exercise sheets
  - Equal conditions for Java and C++ people
  - One more tutor than last time + encouraged feedback on code

Course evaluation results 9/9

- Planned improvements for next time
  - Even more and more concrete implementation advice
    - We have assembled a long list of (mostly small, some bigger) issues over the course of the semester
  - Try to provide test cases, at least basic ones
  - Provide master solutions, at least for the first exercises
  - Better example for Contraction Hierarchies
  - For more complex drawings, prepare only the outline and complete it **live** during the lecture
  - Better submission system for the course evaluation

Exam 1/2

- Exam date is **Monday, August 20** at **2:00 pm**
  - Will take place in HS 026 + it will last (only) 90 minutes
  - There will be **4 tasks**, out of which you can select **3 tasks**
  - Three kinds of tasks are possible
    - Execute an algorithm from the lecture, or some variant of it, on a given example (on paper)
    - Write a small program to solve a variant of a problem we have seen in the lecture
    - Compute, reason about, or prove a non-trivial (but also not very difficult) property of an algorithm or data structure from the lecture, or some variant of it
  - Let's have a look at the exam from last year ... it's on the **Wiki**

Exam 2/2

- The **bachelor students** (and only those)
  - ... must take an **oral** exam
  - There are only two bachelor students this year
  - So let's wait for the exam registration deadline (end of this week) to be over ...

    And then fix a date by person communication (mail)
- For **all**
  - There will be a subforum for asking questions concerning the exam (while you prepare for it)

Work in my group 1/4

- Chair for Algorithms and Data Structures
  - Our work roughly subdivides as
    - 1/3 theory (new algorithms, complexity analysis, etc.)
    - 1/3 algorithm engineering (efficient implementations)
    - 1/3 software engineering (good, durable software)
  - Current projects
    - Route planning ... next slides
    - Search engines, in particular: CompleteSearch & Broccoli
  - Current readings
    - http://ad.informatik.uni-freiburg.de/papers

Work in my group 2/4

- Multi-modal route planning
  - We have learned in this course: routing on **road** and **transit** networks are two different problems
  - Yet: combining the two is highly desirable:
    - driving, walking, bus&train, bike, flights
    - the routing system should consider (almost) all possible combinations and propose the best one**s**
  - In particular: multi-modal routing for **Freiburg**
    - in collaboration with the local transit agency VAG

Work in my group 3/4

- Real-time updates
  - In the lecture we have considered **static** networks
    - that is, travel times / schedules are fixed
  - In real life you have **traffic jams** and **delays**
  - We have learned that for fast query times you need precomputation, which takes some time
  - **Question:** how to accomodate real-time updates without having to do the precomputation from scratch again
  - In particular, for the transfer patterns algorithm:
    - Enough to **only** recompute the direct-connection data structure (fast), and **not** the transfer patterns (slow) **?**

Work in my group 4/4

- Result diversity
  - Users want a variety of results, especially for transit
    - trips at **different times**
    - but also **different routes**
  - Multi-criteria gives you some diversity, but not all
  - (Slightly) non-optimal solutions can also be interesting
  - But there is also a lot of diversity that is **not** interesting
    - travel time: 2h12min; walking: 750m versus
    - travel time: 2h10min; walking: 770m (similar route)
  - **Problem 1:** good definition of diversity that users like
  - **Problem 2:** compute such result sets efficiently