Overview of this lecture

Organizational

- Your experiences with ES13
- Official evaluation
- Infos about the exam
- Work at our chair

SPARQL, SQL, SQLite
results + discussion
when, where, how, tasks
how, projects, next courses
Experiences with ES13  1/2

Summary / excerpts

- Interesting + fun exercise + quick, once the idea was clear

- Problems with TABs in doctest

  Simple fix from the forum: use `r''' ... '''` instead of `''' ... '''`

- Confusion for some, because different tables in lecture than in exercise sheet (2 vs. 3 columns, different predicate names)

  I understand, but it was deliberate, so that you could not just copy what I did in the lecture → deeper understanding needed
Experiences with ES13  2/2

Results

- Indices are essential for sub-second query times
  *Without indices, first query from ES13 takes > 20 minutes*

- Join order makes a big difference in query time
  *Like in the example in Lecture 13*

- Strange information in Freebase: many people said to have died from natural cause and murder at the same time
  *Actually, that's a problem we introduced in the process of "simplifying" the data for you (the original Freebase data is more complex than what we gave you)*
Participants

- Registered for exam: **87** … record number
  
  51 x MSc Inf, 24 x BSc Inf, 6 x M.Sc. ESE, 6 x Other

- Participated in the evaluation: **61** … thanks!

  That's 71% of the number of registrations, compared to over 90% in previous year (where half of the points were necessary to be admitted to the exam)

- Nominations for teaching award: **40** … thanks again!

- In the following, a summary of your feedback

- You find all the details [linked on the course Wiki](#)
Results Course Evaluation

Style of the course

- Learned a lot: 69% fully agree, 28% agree, 3% ok
- Well explained: 74% fully agree, 20% agree, 6% ok
- Asks own activity: 71% fully agree, 20% agree, 9% ok
- Level of contents: 46% appropriate, 54% high, 0% low
- Overall mark: 67% very good, 28% good, 5% ok

- You liked: interesting contents, practically useful, well structured, well prepared, complex stuff explained in an understandable manner, good examples, live coding, enthusiasm, great support on forum, great recordings, …

- Criticism / suggestions: see slide 9
Results Course Evaluation  3/7

■ Student's effort

- Effort relative to ECTS  ...  1 = very high, 5 = very low

  19% x 1  32% x 2  48% x 3  2% x 4  0% x 5  this course

  16% x 1  29% x 2  50% x 3  5% x 4  1% x 5  department average

- Your time investment  ...  1 = very high, 5 = very low

  30% x 1  32% x 2  27% x 3  11% x 4  0% x 5  this course

  24% x 1  28% x 2  25% x 3  15% x 4  8% x 5  department average

My feeling: about half of you participated very actively, the other half not so much + more critical than last time
Results Course Evaluation  4/7

Materials / Online Support / Tutors

- Helpful: 82% fully agree, 15% agree, 3% ok
- Consumed lecture by presence or by video recordings:
  26% presence, 30% recordings, 43% both ways
  51% presence, 15% recordings, 27% both ways
- Video recordings (Frank + Dennis): great quality, thanks!
- Assistant (Elmar): great support for sheets and in forum
- Tutors (Björn, Claudius, Sabine, Simon): fast feedback, very friendly, extensive feedback much appreciated (in most cases), a few complaints about opaque grading
Results Course Evaluation  5/7

- Criticism / Suggestions
  - Provide crash course or cheat sheet for Python
  - Room too small / lack of oxygen
  - Mistakes in the provided code + more explanations please
  - More mathematical details
  - More about: SPARQL, reasoning, information extraction, …
  - More introduction to linear algebra
  - Web apps: some loved it, others preferred other topics
  - Physical tutorial would be nice from time to time
  - Easier exam please
Results Course Evaluation  6/7

- Planned improvements from WS 13/14 course
  - Fix mistakes on slides ... without introducing new ones
  - Further reduce time effort for exercise sheets
  - Forever work on time management
  - Different room ... HS 036 was terrible
  - Get slides on Z-Test and T-Test right
  - I took extensive notes for each lecture on what to improve

If you look at the slides of this WS 15/16 course, you will see quite extensive changes (in structure and in contents) compared to the WS 13/14 course
Planned improvements for WS 16/17 course

- Try to get HS 026 (if there are again so many people)
- Provide DIY tutorial on SVN + Jenkins in English
- Provide Python crash course and/or cheat sheet
- Provide live tutorial every four weeks or so + say early in the course that live meetings with tutor possible anytime
- More mathematical background for those who want it
- Even better error checking of exercise sheets
- Further optimize time management
- Consider the extensive new notes I took for each lecture
Where, when, what to bring

- Oral exams (B.Sc. Computer Science students only):
  
  Wednesday – Friday, February 24 – 26, 30min in 51-02-28

- Written exam (all other participants):
  
  Tuesday, February 23, 2:00 – 4:00pm in HS 026+036

- Please bring: **student id, colored pens, brain**
  
  student id: make sure you look like your photo or vice versa
  
  colors: greatly improve readability for examples / drawings
  
  brain: greatly improves quality of answers in general
Type of exam

- The written exam is **open book**

  That is, you can bring books, paper, etc … but please be ecological when printing out slides + good for your karma

  Electronic devices of any kind are obviously not allowed

- In the oral exam, ask us if you are missing some detail

  No time to start understanding things during the exam …
  this also holds true for the written exam

- There will be a sub-forum for questions about the exam

  Don't ask all your questions in the night before the exam
Types of questions

- **Type 1:** Do algorithm (or variant thereof) by example, like we often did in the lecture … see colored pens

- **Type 2:** Implement a small program (in Python, Java or C++) … small indeed, max 10 lines per task!

- **Type 3:** Small calculations or proofs … see brain

- In general: the emphasis is on (basic) understanding, not on learning things by heart

- Important: all the contents + insights from the exercises is (very) relevant for the exam
Preparation

- To check whether you understood something:

  Put away the material from the lecture and try to write it down / prove it **in your own words / formalism**

  There is no point in learning the individual steps of a proof or argument by heart … it doesn’t work that way

- Once the basic stuff from the lecture is understood, the best preparation is to do all the exercise sheets

  If you work your way through all the exercise sheets (yourself), there is no way you can fail the exam

- Also: work through the old exams (linked on the Wiki)
Task 1 from WS 13/14 exam

1.2 \( M > 8 \Rightarrow 4 \text{ Bits for remainder, 1 Bit for leading 1} \)
\( \Rightarrow 5 \text{ Bits per Code minimum} \)

1.3 \[
\text{vector< bool > gallum (int x) } \\
\text{vector< bool > code;} \\
g = x / M; \quad \text{// quotient} \\
\text{r = x } \% \text{ M; } \quad \text{// remainder} \\
\text{for (int i = 0; i < g; i++) } \text{ code.push_back(0);} \\
\text{code.push_back(1);} \\
\text{code.push_back(r & 1 ? 1 : 0);} \\
\text{code.push_back(r & 2 ? 1 : 0);} \\
\text{code.push_back(r & 4 ? 1 : 0);} \\
\text{return code;} \\
\]

1.4 \[
+1 : 1001 \\
+3 : 01001 \\
\Rightarrow 1001010011001... \\
\]

1.5 
\[
P_1 = P_2 = \frac{1}{2} \\
H(X) = \frac{1}{2} \cdot \log_2 12 \cdot 2 \\
\text{average code length } \]

\[
q = x \div M \quad r = x \mod M \\
q = 0 \Rightarrow m \quad \text{binary} \\
\]
Task 2 from WS 13/14 exam

1.1
\[ x = \text{angela} \quad y = \text{angelia} \]
\[ \text{PED}(x, y) = \text{ED}(\text{angela}, \text{angelia}) = 1 \]
\[ \text{PED}(y, x) = \text{ED}(\text{angelia}, \text{angela}) = 2 \]

2-grams of (x):
\[ \{ \text{am}, \text{ng}, \text{ge}, \text{el}, \text{la} \} \]

2-grams of (y):
\[ \{ \text{am}, \text{ng}, \text{ge}, \text{el}, \text{di}, \text{ni}, \text{ma} \} \]

1.2
\[ \text{comm}_2(x, y) = 4 \]

1.3
\[ |2 \text{-grams}(x)| = |x| - 1 \]
\[ \Rightarrow \exists y' : \text{ED}(x, y') = 1 \]
\[ \text{PED}(x, y) = 1 \]

If \( x = y' \), then \( \text{comm}_2(x, y') = |x| - 1 \)

Some ED operations (delete) can destroy at most 2 2-grams

1.4
Example for \( |x| = 3 \):
\( \text{PED}(x, y) = 1 \)
\( \text{comm}_2(x, y) = 0 \)
Task 4 from WS 12/13 exam

\[
\begin{pmatrix}
1 & 1 & 0 & 1 & 0.5 \\
1 & 0 & 1 & 2 & 1 \\
0 & 0 & 1 & 1 & 0.5
\end{pmatrix}
\rightarrow
\begin{pmatrix}
1 & 1 & 0 & 1 & 0.5 \\
1 & 1 & 1 & 2 & 1 \\
0 & 0 & 1 & 1 & 0.5
\end{pmatrix}
= \begin{pmatrix}
1 & 0 \\
1 & 1 \\
0 & 1
\end{pmatrix}
\cdot \begin{pmatrix}
1 & 1 & 0 & 1 & 0.5 \\
0 & 0 & 1 & 1 & 0.5
\end{pmatrix}
\]

\(\text{rank} = 3\)

\(\text{rank} = 2\)
How we work

- We solve practically relevant problems
  Route Planning on Google Maps, Search As You Type, Semantic Full-Text Search, Question Answering, …

- We make our software + results available to the public
  This requires an effort to write good software, good documentation, nice user interfaces, and so on …

- We use theory as a (vital) tool, not for the sake of theory
  Without a solid theoretical understanding, solving complex problems remains hacking and guesswork
Work at our Chair  2/5

- Supervision
  - Similarly as in the lecture:
    
    Very good infrastructure + support, but apart from that you are supposed to work independently
    
    Great for enthusiastic people who care about practical stuff and who want to get things done
Machine Learning

- We are building more and more on **machine learning** to solve our problems

  Not because it's fashionable … but because it's practical

  It's quite obvious that learning is the future for problems like natural language understanding

- You have seen a few learning algorithms in this lecture

  The complexity lies not so much in the algorithms, but in understanding how and why they work how well
Current projects

- Semantic search
  Search with "understanding" of the query and documents
  Show demos of Broccoli and Question Answering

- Route planning
  We are behind the algorithm on Google Maps (Transit)
  Good models, efficient algorithms, a working system

- Other projects
  Check out the demos on our webpage
Upcoming courses

- **Programming in C++** … in the SS 2016
  
  2nd semester B.Sc. Info + 4th semester B.Sc. ESE

- **Information Retrieval** … in the WS 2016/2017
  
  You know it … become a tutor if you do a great exam!

- **Algorithms and Data Structures** … in the SS 2017
  
  Basic course for 2nd semester B.Sc. Informatik students

- **B.Sc. / M.Sc. projects or theses**

  Offered all the time, just ask

  Ask again if no response, and again, and again … please!