University of Augsburg, Institute of Computer Science
Prof. Dr. P. Fischer
J. Kastner, L. Rudenko

SS 2019
26. Apr. 2019
Sheet 1

Analyzing Massive Data Sets

Exercise 1: Bonferroni's Principle (live)

In the first lecture **Bonferronis's principle** was mentioned: Data mining runs into the risk of "discovering" meaningless patterns. For "suspect identification" we want to find (unrelated) people who, on two different days, were both at the same hotel (slide 25). Hotels can be different on each of days.

We make the following assumptions:

- 10^9 people are being tracked
- 10^3 days observation period
- Each person stays in a hotel 1% of time (1 day out of 100)
- Hotels hold 100 people. Hence, there are 10^5 hotels - enough to hold the 1% of a billion people who visit a hotel on any given day.

If everyone behaves randomly, the expected number of "suspicious" pairs of people is 250000.

a) Verify this result on your own. Describe your modelling and show the computation.

b) Calculate the number of suspected pairs if the following changes were made to the data **at the same time**:

   - The number of observation days was raised to 2000.
   - The number of people observed was raised to 2 billion (and there were therefore 200000 hotels).
   - We only reported a pair as suspect if they were at the same hotel at the same time on three different, not necesarily adjacent days.

Exercise 2: Python - Basics (homework)

For the Python exercises we recommend **Anaconda 5.1** Python 3.6 edition: a standalone, feature-complete Python distribution available for the common OS platforms:

https://docs.anaconda.com/anaconda/install/

You may also use other distributions and tools, but feedback from our side is more difficult.

In the lecture a number of core concepts were presented. In this exercise you should try to implement a number of algorithms in Python.

a) Implement the **Merge Sort** algorithm in Python.
You can find a description of Merge Sort at https://en.wikipedia.org/wiki/Merge_sort.

b) Implement **Depth-first Search** in Python.
   Short description of Depth-first Search: https://en.wikipedia.org/wiki/Depth-first_search.

c) Implement **Breadth-first Search** in Python.
   Breadth-first Search in Wikipedia: https://en.wikipedia.org/wiki/Breadth-first_search.

Exercise 3: Python - Pandas (live)

In this exercise we will perform some basic data analyzing with Pandas. As datasets we use a part of TPC-H benchmark. The relations used for this exercise (*supplier, part, partsupp*) can be downloaded from Megastore: https://megastore.uni-augsburg.de/get/4PaGJ9OCdX/

a) Find 25 suppliers with the lowest account balance.

b) How many suppliers have a positive account balance?

c) Find out all brands produced by the same manufacturer and calculate the items number and the total sales price for each brand of each manufacturer.

d) How many items have 3 words in their name?

e) How many different items does each supplier have?