# Welcome to ACS TA session 6

Alexander, Lasse, Svend, Yiwen, and Yuan

Department of computer science, University of Copenhagen

Academic year 2018-2019, Block 2

# Agenda for today

**Feedback on Theory Assignment 2**

**Data processing**

- Multi-way external merge sort exercises

# Theory Assignment 2 – Q1&Q3

**T1:** SA(a)**R(a)**XA(b)SR(a)                    **R(b)**XR(b)    **C**
**T2:**                    XA(a)**W(a)**XR(a)   **C**

Schedule like this can be generated by 2PL scheduler!

**Ta**: R(a)                                        **W**(b)     **C**
**Tb**:        R(c)R(b)**W**(c)**W**(b) **C**
**Tc**:                                R(c)**W**(c) **C**

- **Tb** completes before **Ta** begins its write phase. Since the intersection of write set of **Tb** with the read set of **Ta** is empty, validation of **Tb** would succeed.
- **Tb** completes before **Tc** begins and that **Ta** and **Tc** access no data items in common, so validation of **Tc** must also succeed.

# Theory Assignment 2 – Q2

- Volatile = Main Memory = RAM

- Non-Volatile = Disk, such as a Harddrive / USB

- Stable Storage = Concept of a storage system.

# Theory Assignment 2 – Q4&Q5

- **Analysis phase**
  - Remember STATUS column for transaction table.
  - Committing transactions may not flush corresponding dirty pages to disk immediately due to no-force policy. Thus, dirty pages of a committed transaction should be still kept in dirty page table.
- **Redo phase**
  - Do NOT log redo actions.
- **Undo phase**
  - Add abort record for each transaction marked as running first.
  - Undo aborted transactions in reverse order of LSNs.
  - UndoNextLSN is needed for CLR records.
  - When undoNextLSN is NULL, add an end record for that transaction.
- **Memory resident database**
  - Corresponds to no-force no-steal system.

# Exercise 1 - multi-way external merge sort

Consider a relation R consisting of 10.000.000 tuples. One field is the sort key field, which we consider is unique. Tuples of R are divided into disk blocks of 16.384 bytes per block. 100 records are fitted into one block. The machine on which the sorting happens has one disk and 100 megabytes of main memory available.

1. What are the phases in multi-way external merge sort?

2. How many passes are needed to sort the relation R?

3. By using the tournament replacement sort, how does it affect the length of a run, in best case and in worst case?

4. Assuming a block read/write takes 11 ms, how long does it take to sort R?

# Exercise 2 - multi-way external merge sort

Consider the following sequence of numbers:

67 12 45 84 58 29 76 7 91 81 39 22

1. Assuming a memory size of 3 pages where 1 page can store only one of the records in the list, show the steps of implementing a multi-way external merge sort.

2. How many passes are required?

# Thank you

Alexander, Lasse, Svend, Yiwen, and Yuan

Department of computer science, University of Copenhagen

Academic year 2018-2019, Block 2