COMP207 Tutorial Exercises
Week 6 (9^{th}–11^{th} November)

The exercises below provide the opportunity to practice the concepts and methods discussed during this week's videos, i.e. the ones on query processing.

If you rather want to do something more practical to do with optimization AND B+ trees/indices and have access to a computer, you can instead try to do the tutorial on how to use index in MySQL `https://www.mysqltutorial.org/mysql-index/mysql-create-index/` (the optimization part comes from it showing you how to see how MySQL have optimized queries, i.e. using the EXPLAIN command in front of queries). You will need to use the sample database from `https://www.mysqltutorial.org/mysql-sample-database.aspx` to do that tutorial. It gives you a zip file containing just an sql file that creates the sample database. You need to run the sql file either using the command line or the Workbench, after connecting.

You can also try, now or when you next have access to the internet, to play around with the online visualization tool for B+ trees on `https://www.cs.usfca.edu/~galles/visualization/BPlusTree.html`. The references for the lectures on the Canvas course page provide some entry points.

Solutions to most of the questions will be be provided Friday like normal, except for the last one, which would be too long to give a solution for during a video.

Query Plans

As you know from this tutorial and the videos, a DBMS takes SQL queries as input and translates these into *query plans* that are not yet optimised. Such plans can be represented as relational algebra expressions or as trees. This exercise is about the non-optimization part.

Exercise 1. Represent the following relational algebra expressions as query plans (in the form of a tree):

(a) $\pi_{B,D}(R \bowtie \rho_{B \to D}(S))$

(b) $\rho_{C \to D}(\sigma_{A=a \text{ AND } C=b}(S)) \bowtie (\pi_{A,C}(S) \times \rho_{A \to D, C \to E}(R))$

Let us now assume a database with the following two relations:

R

A	B	C
a	a	b
b	d	a

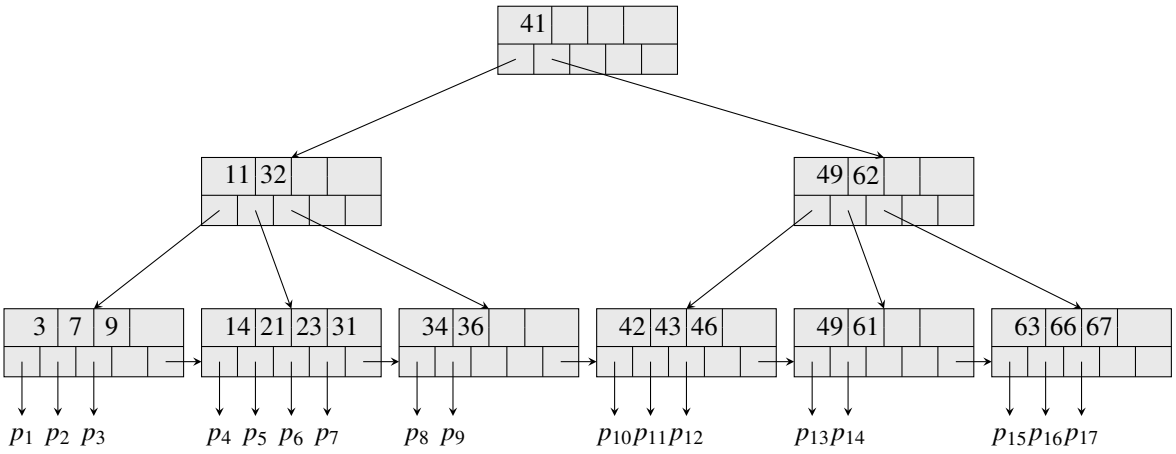
S

A	B	C
a	d	b
a	c	b
d	c	a

Exercise 2. Execute your query plans from Exercise 1 step by step on this database, as done in the examples and exercises in the video "Executing a query plan".

B+ Trees

Consider the following B+ tree:



Exercise 3. Use the method from the page "Idea, searching and inserting in B+-tree" to look up the pointers associated with the following values in the B+ tree. Try to follow the method step by step.

(a) 61

(b) 62

(c) All values greater than or equal to 44

(d) All values in the range from 47 to 64 (i.e., all values x with $47 \leq x \leq 64$)

Exercise 4. Insert the following value/pointer pairs into the above B+ tree:

(a) Value: 41; pointer: q

(b) Value: 60; pointer: r

Exercise 5. Insert the following value/pointer pairs into the B+ tree above:

(a) Value 22; pointer p

(b) Value 17; pointer p (no solution provided)

Implementing B+ Trees

Exercise 6 (no solutions provided)**.** Implement the B+ tree data structure in a programming language of your choice. Then try to experiment with your implementation by building a B+ tree for a large collection of values, by looking up values, by performing range searches, by inserting values, etc.