Implementation of DBMS
Exercise Sheet 2, Solutions
Klingemann, WS 2024 / 2025

1) Suppose you use Two-Phase Multiway Merge Sort in the scenario described in the lecture slides. Tell how many disk I/O's are needed for the sort if the following changes are applied:
a) The number of records of the file is doubled.
Solution: We should first check, whether two passes are still sufficient for the increased number of records. We have calculated in the lecture that $6.71 * 10^9$ records are possible so that this is the case. Second, we know that with two passes we have 4 I/O's per block. As twice the number of records require twice the number of blocks, we therefore have twice the number of I/O's, i.e. 2,000,000 I/O's.
b) The size of blocks is doubled, to 8192 bytes.
Solution: When we double the size of the blocks, we can store twice the number of records in a block. Thus, we need half the number of blocks and therefore half the number of I/O's, i.e., 500,000 I/O's. (However, as the blocks are larger, we also increase the time for an I/O!)

2) Suppose we have a relation whose n tuples each require R bytes, and we have a machine whose main memory M and disk-block-size are just sufficient to sort the n tuples using Two-Phase Multiway Merge Sort. How would the maximum n change if we made one of the following modifications of parameters?
a) Double B
b) Double R
c) Double M
Solution: We know that the maximum number of records is determined by $M^2 / (R * B)$. As a result, for a) and b) we get half the number of records and for c) four times the number of records.

3) You want to use the Two-Phase Multiway Merge Sort (or its extension to a different number of phases as appropriate) to sort a file. The file consists of 119,985 records. Each block can contain 20 records. We have 10 main memory blocks available.
a) How many blocks do we need to store the file if each block is as full as possible?
Solution:
We need $\lceil$(119,985 records) / (20 records/block)$\rceil$ = 6000 blocks. Note, that the last block is not completely full but nevertheless needed to store all the records!

b) How many phases do we need?
c) What is the required number of I/O's?
d) How many sorted sublists do we have after each phase?
Solutions:
Phase 1: We get $\lceil$(6000 blocks) / (10 blocks/sublist)$\rceil$ = 600 sublists
Phase 2: We get $\lceil$(600 sublists) / (9 sublists merged to 1)$\rceil$ = 67 sublists
Phase 3: We get $\lceil$(67 sublists) / (9 sublists merged to 1)$\rceil$ = 8 sublists
Phase 4: We get $\lceil$(8 sublists) / (9 sublists merged to 1)$\rceil$ = 1 sublists, i.e., we have created the sorted file.
As a result, we needed 4 phases. This implies that we need 8 I/O's per block and therefore in total (8 I/O's per block) * (6000 blocks) = 48000 I/O's.

4) Consider the following relations:

R:

A	**B**
a	b
c	b
d	e

S:

B	**C**
b	c
f	a
b	d

Calculate the following relations:

a) $\pi_A(R)$

A
a
c
d

b) $\sigma_{A=d}(R)$

A	**B**
d	e

c) $\pi_A(\sigma_{B=b}(R))$

A
a
c

d) $R \times S$

A	**R.B**	**S.B**	**C**
a	b	b	c
a	b	f	a
a	b	b	d
c	b	b	c
c	b	f	a
c	b	b	d
d	e	b	c
d	e	f	a
d	e	b	d

e) $R \bowtie S$

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

f) $R \bowtie_{A=C} S$

A	R.B	S.B	C
a	b	f	a
c	b	b	c
d	e	b	d

g) $\pi_A(R \bowtie S)$

A
a
c

h) $\pi_A(R) \bowtie S$

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

why this? and there is no common attribute, so it shall be

if no common attributes, then like cartetian, all other atrib