**Database management, spring 2002** **Exercise 4 (22. –26.4.)**

**For active attendance: min 3 tasks done A task marked with (**) is counted as two tasks.**

1. For the file of tasks 1 and 2 on the last week, estimate the size and the number of disk accesses of two dynamic hash indexes: extendible hashing (E&N, 145-146) and dynamic hashing. The latter is not included in E&N, 3rd edition. Its picture can be found on page 52 of the chapter 3 in the (Finnish) lecture material or in E&N's older version (E&N2, p. 93). (The 'dynamic hashing' case is optional in this task; the principle will be explained in the exercise session.)

2. Compare the query trees given in Fig. 18.4 (E&N) with the following relation sizes: employee: 5000, department: 50, and project: 500 rows. (The implementation of the relations and indexes will not be covered now.)
a) What are the sizes of the intermediate results in the initial tree (b)? The tree can be slightly modified by using different orders for the Cartesian products (based on associativeness); the same question for these modified trees?
b) Three optimizations separate tree (a) from tree (b): the early selection, and two changes of Cartesian products to joins. What is the effect of every single optimization to the intermediate results?
c) When should the projections be done in this example?

3. Construct a small example to show that the projection operation commutes neither with the intersection operation nor with the difference operation. (Cf. E&N, p. 612, point 11. This is not a difficult task even if some textbooks have claimed those to commute!)

4. Consider the following query

```
select e.FNAME, e.LNAME, n.DEPENDENT_NAME
from EMPLOYEE e, DEPARTMENT d, DEPENDENT n
where e.SSN = n.ESSN and e.SSN = d.MGRSSN and d.DNAME = 'Research'
and n.RELATIONSHIP='SON'.
```

a) What is the meaning of the query?
b) Give the projection-selection-product relational expression directly corresponding to the query. Also give the corresponding query tree. Associate the cartesian products in such a way that presumably will result in the best optimization effect.
c) Optimize the query using heuristics "apply the selections as early as possible". Give the query tree and the corresponding relational expression for the optimized query.
d) Is it possible to optimize the query still further?

5. a) How would you evaluate the query of the previous problem when there is an index on attribute SSN to relation EMPLOYEE and an index on attribute DNAME to relation DEPARTMENT? Are there differences between indexes of different kind?
b) Is it possible to make the query more efficient using some other index?
c) Is it reasonable to maintain indexes for the relations of this query – compared to the other relations of the company database?

6. (**) Let us assume the query

```
select FNAME, LNAME, PNO, HOURS from EMPLOYEE, WORKS_ON
where ESSN=SSN and HOURS > 20;
```

Evaluate the number of disk accesses (minimum and maximum values) needed when either the nested-loop join or single-loop join is used. (cf. E&N, p. 594.) The relation EMPLOYEE has 10000 rows of 100 characters, and the relation WORKS_ON 30000 rows of 20 charcaters. The page size is 2KB.