

# Problem sessions and hand-in

Advanced database technology

February 20, 2006

1. Consider an SQL statement such as `SELECT * FROM Students WHERE grade=8`; If there is a secondary index on `Students` the DBMS may choose to use it to find the relevant tuples (an “index scan”) or to just go through the whole relation (a “full table scan”). In the database tuning literature you can find rules of thumb such as “*A full table scan is generally faster than an index scan if more than 20% of the rows need be examined.*”.
  - (a) Why is it not always an advantage to use the secondary index?
  - (b) What are the factors that influence whether it saves time to use an index for a given query?
2. Suppose we want to search among  $N$  keys, that internal memory can hold  $M$  keys/pointers, and that a disk block can hold  $n$  keys/pointers. Further, suppose that the *only way* of accessing disk blocks is by following pointers. Show that a search takes at least  $\lceil \log_n(N/M) \rceil$  I/Os in the worst case. **Hint:** Consider the size of the set of blocks that can be accessed in at most  $t$  I/Os.

## To be handed in at the latest March 6:

1. Exercise 13.2.6 from G UW
2. Problem 1 from the ADBT exam, June 2004.

The exam problems can be downloaded from the course home page.