



Exercise Sheet 3: Indexing and Access Paths

Exercise 1

- What is minimized by the buffer manager?
- What is important for a sequential file organization to be efficient for a (sequential) scan of the file?
- What index is preferable for range queries: primary index or secondary index?
- Imagine you want to store a huge image in a database. What datatype is usually used? And what is the impact regarding the block size?

Exercise 2

A file has $r = 20,000$ *client* records of fixed length. Each record has following fields: UID (9 bytes, unique), Name (40 bytes), Address (80 bytes), Email (40 bytes), Phone (12 bytes), Birthrate (8 bytes), and Sex (1 byte). An additional byte is used as deletion marker. The DBMS file block size is 512 bytes.

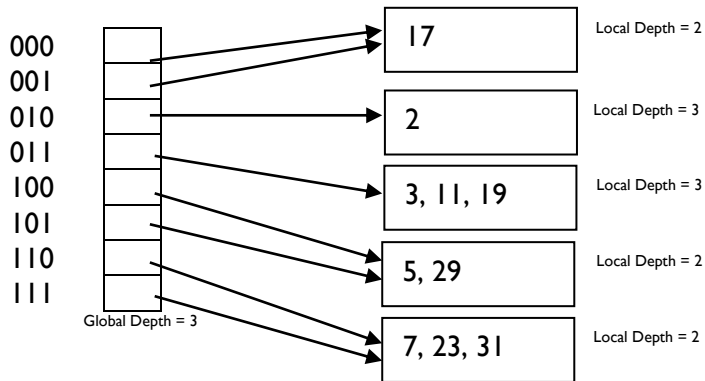
- Calculate the record size of a single record.
- How many blocks are necessary assuming a spanned block organization ignoring the pointer size? How many are necessary for an unspanned organization?
- How many block accesses assuming an unspanned organization are necessary to locate a random record using linear search on average and in the worst case?
- Assuming the records are ordered by their UID. What is the maximum and average number of block accesses necessary to locate a record by UID using binary search again assuming an unspanned organization?

Exercise 3

- What are the differences among **primary**, **secondary**, and **clustering** indexes? How do these differences affect the ways in which the indexes are implemented? Which indexes are **dense**, and which are not?
- Is it possible in general to have two primary indexes on the same relation for different search keys? Explain your answer.
- How does multilevel indexing improve the efficiency of searching an index file?
- What general requirements should a hash-function fulfill?

Exercise 4

Given is the following extendable hash table which uses the hash function $h(x) = x \bmod 8$ and a bucket size of 3:



Show the hash table after each of the following operations (applied in that order):

- Insert 1
- Insert 46
- Delete 11
- Delete 19