Exam preparation topics
CSC 443. Database Management Systems

Part I. Handling large amount of data efficiently

1. Storage media
   What are advantages and disadvantages of storing data in RAM? on magnetic disks? On tapes?
   Main components of disk latency.
   Why random access to disk data is significantly slower than sequential access?
   What is a key to lower I/O costs?

2. Buffering
   How the buffer pool is organized?
   What is a dirty page? What is a dirty bit?
   What is a pin count? Why pin count need to be a number and not a single bit?
   What pages are written back to disk?
   Replacement policies: First In-First Out, Least Recently Used*, Most Recently Used*, Clock*, the second chance algorithm*.
   What is sequential flooding? Sample code for sequential flooding. What are the solutions to sequential flooding problem?
   What is double-buffering?

3. Algorithms
   I/O model of computation. What is beneficial for algorithms which work on inputs that cannot fit into RAM?
   Two-Phase Multiway Merge Sort* (2PMMS). Phase I. Phase II. What is in input buffers? What is in an output buffer? How many buffers max for RAM of size M? Max number of records that we can sort in 2 passes. Given N – number of records – determine if you can sort them in 2 passes with the memory of size M and a block of size B. How to sort larger relations? How many records we can sort with 3 passes?
   Algorithms that benefit from sorting.
   Software and hardware methods for improving performance of 2PMMS.
   How to incorporate duplicate elimination (or other operations) into the merge phase of 2PMMS.
   Design similar algorithms to solve problems such as: set union, intersection, difference etc.

4. Data structures
   Primary and secondary indexes. Dense and sparse indexes. Static trees. Static hashes.

* For these questions you should understand how the algorithm works, be able execute steps of the algorithm, draw the state of all participating data structures etc.
B-tree file organization: clustered files = clustered indexes.
Extendible hashing*, linear hashing*.
When to use B-trees, Hash tables, Grid files, R-trees, Bitmap indexes, inverted indexes.

Part II. Query evaluation

1. Algorithms for implementing relational operators.

2. Logical query optimization


Part III. Map-reduce

   Scalability of systems and algorithms. Recognizing inherently parallelizable tasks.
   Main idea of map-reduce: partitioning by hashing.
   Map-reduce: storage, programming model, data model.
   Problems easily solvable with map-reduce*: word histograms, inverted index. Join in map-reduce.
   Vector-matrix multiplication in map-reduce. Duplicate elimination. Union, intersection, difference.
   Scalability of different parallel architectures.

Part IV. Concurrency

   Transactions. ACID.
   Precedence graphs.
   Enforcing serializability by locks. Two-phase locking. Two main types of locks. Upgrading locks.
   Deadlocks. Three types of locks and their compatibility matrix.

Part V. Recovery

   Preserving data consistency. How to recover from an inconsistent state. Logging.
   Undo log with checkpointing: logging rules, order of disk writes, recovery rules and order.
   Redo log with checkpointing: logging rules, order of disk writes, recovery rules and order.
   Undo/Redo log with checkpointing: logging rules, order of disk writes, recovery rules and order.
   Non-quiescent archiving.
   Coping with disk crashes. RAID 4. RAID 5. RAID 6.

* For these questions you should understand how the algorithm works, be able execute steps of the algorithm, draw the state of all participating data structures etc.