Alg 11/9 R

Mem vs. Disk Mem Fast, Disk slow
10-70 ns vs 3-9 ms

10 year progress Access speeds have doubled
Data storage has increased x 30

Performing A Disc Read

Seek Time → move head to correct cylinder

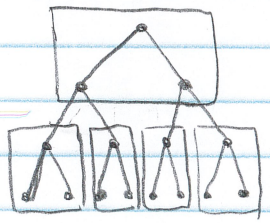
Rotational Latency → Time to reach correct sector by rotation

Transfer time → Time to read a sector of data

If not all together, data read takes ~4 sec
rather than 1/3 sec when data close together

Linear Indexing → Key/pointer pairs to Linear files
- If index too large, do ranges instead of indiv. records

For Trees



Subtrees in same block

2-3 Tree
- Same as BST, but 2 values per node & middle insert
- grows upward
- short, "robust" trees
- never used

Alg 11/9

B-tree → General version of 2-3 tree

What order m for B-tree? → Well first, half of nodes should be full
choose m to minimize resize events

Each node has m children and m values

(!) Very few disk accesses

Search Analysis m=2 $O(\log_2 n)$

+ Insert

+ Delete m $O(\log_{m+1} n \times 2\log_2 m)$

Depth Traversal — Search for next node (Binary search on sorted array)

Why B-tree over Hash? → It's a sorted list that is great for ranges.
→ Hashes don't do ranges well

B+ Trees → Values only stored at bottom
- specifically for querying DB's
- Nodes know siblings and cousins

(++) Can read/page through range of data
- Indexes are solely references
- ½ Full goal

B* Tree → B+ Tree but ⅔ full goal