Learned Indexes

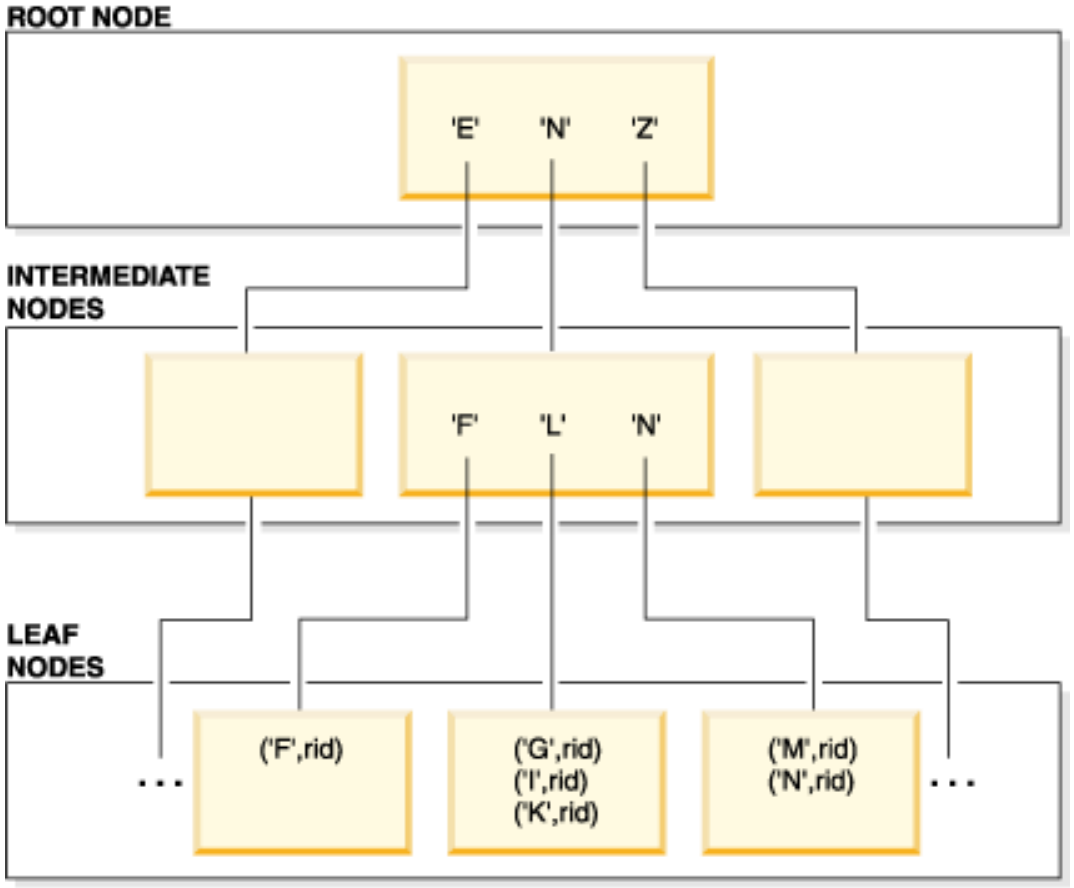
Part II Project - Franz Nowak

What’s a database index again?

- data structure that improves speed of data retrieval operations
- cost of additional writes
- cost of storage space to maintain index
- e.g. b-tree on sorted data: lookup O(log(n)) -> optimal for range lookups

What’s a database index again?

b-tree index

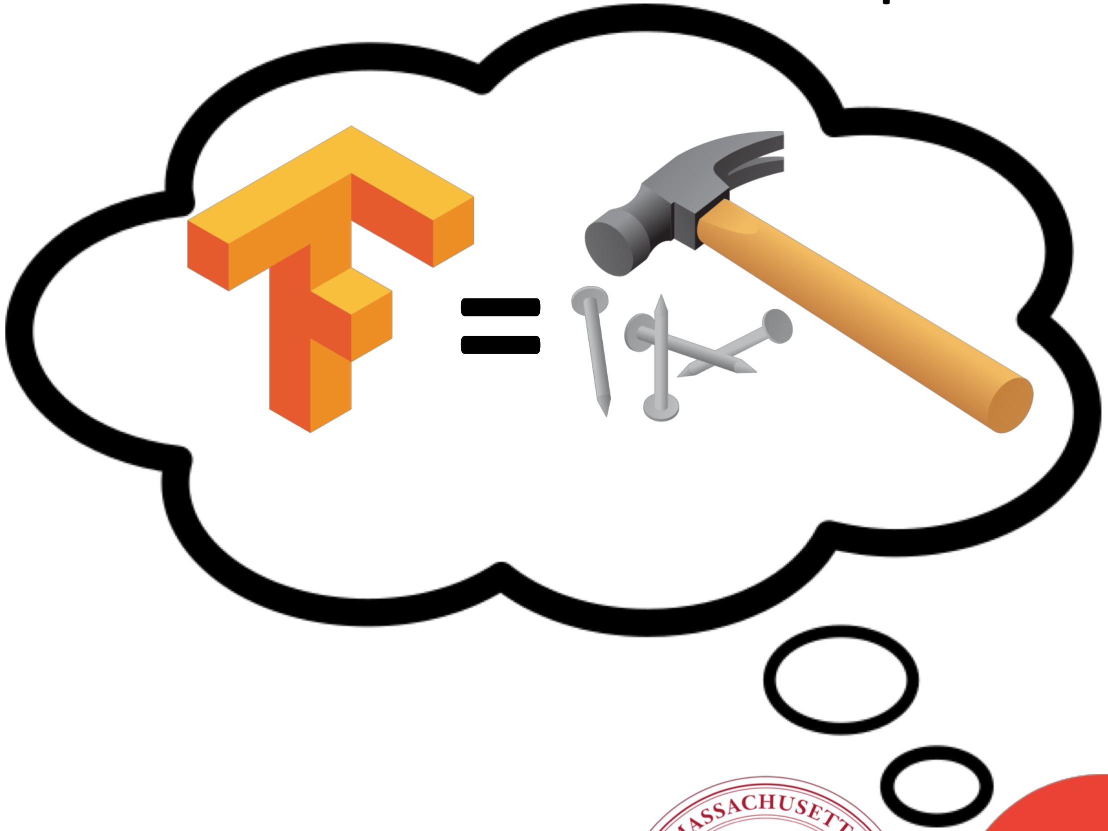


source:https://www.ibm.com/support/knowledgecenter

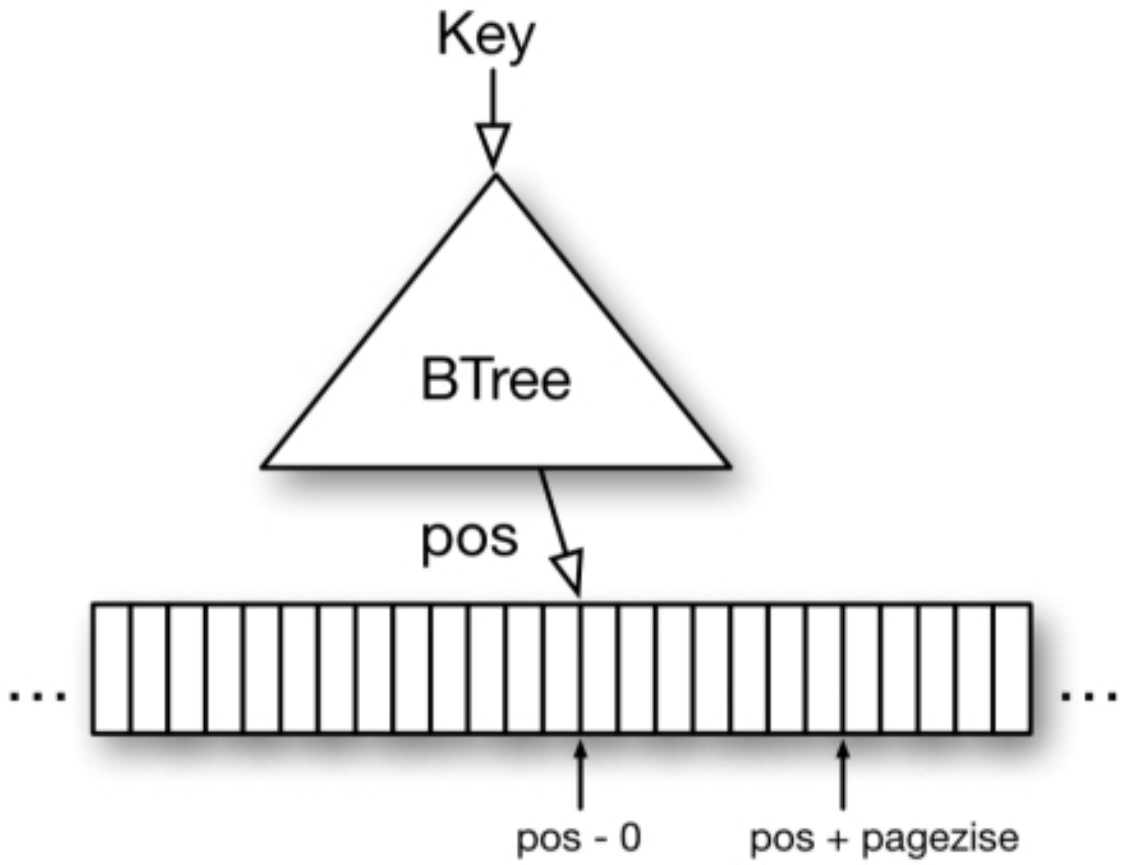
Exploit inherent structure of data

- What if data is all integers from 0 to 1,000,000?
- data_array[lookup_key]
- No need for b-tree, lookup is O(1) time and space!

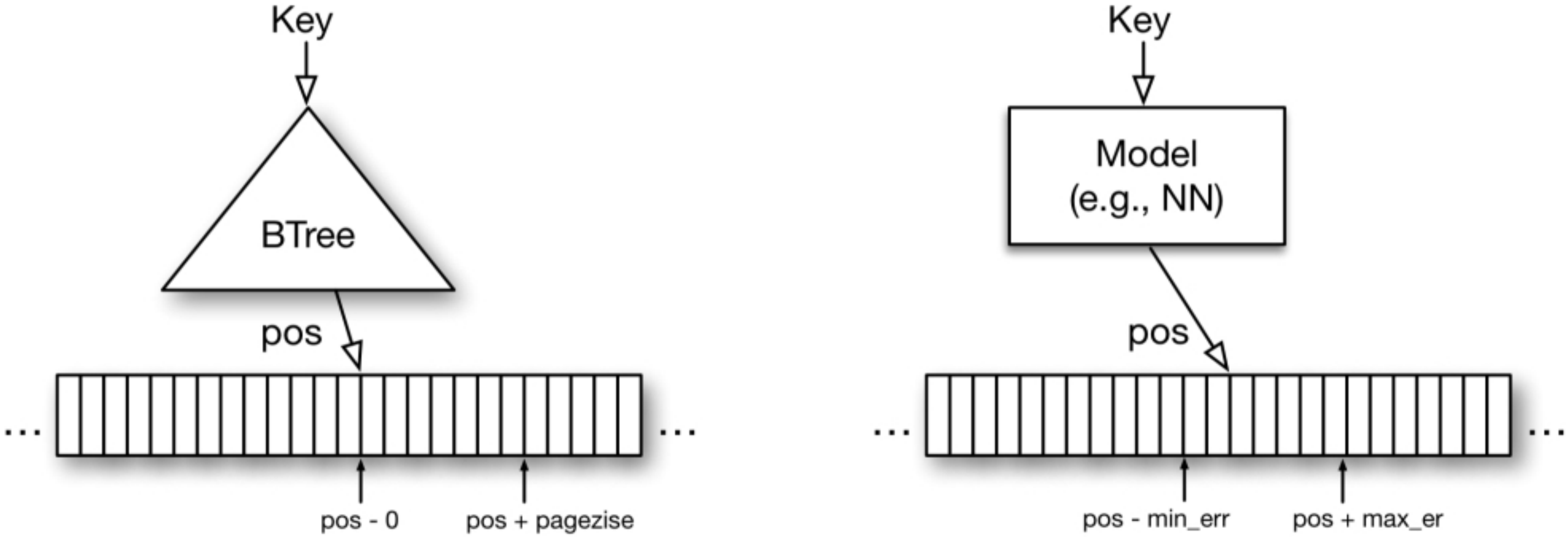
How can we improve on this?



Core idea: B-Trees are models



Key idea: B-Trees are models



To Do: Regression Tree

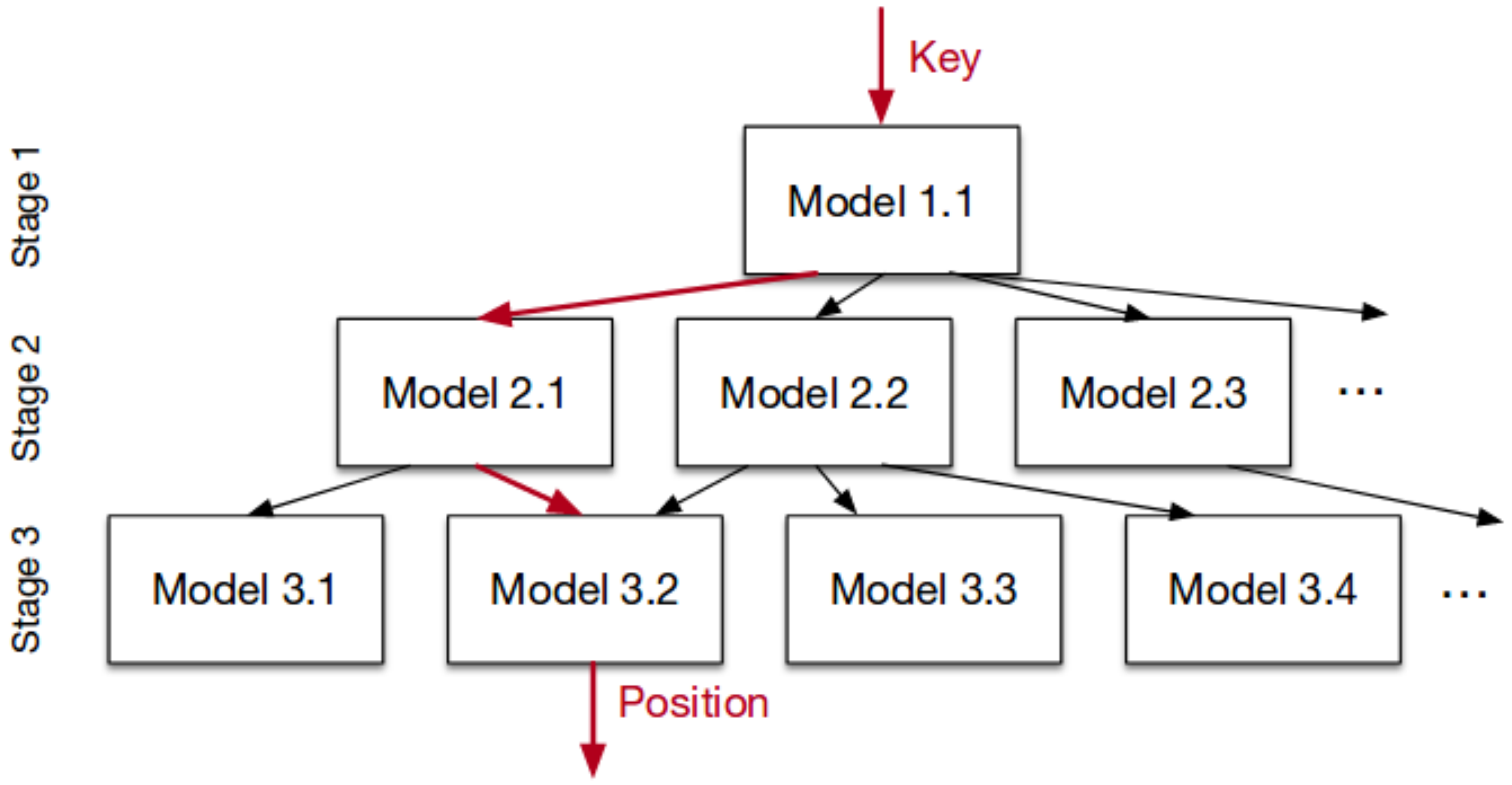


Figure 3: Staged models

Position in sorted array: CDF

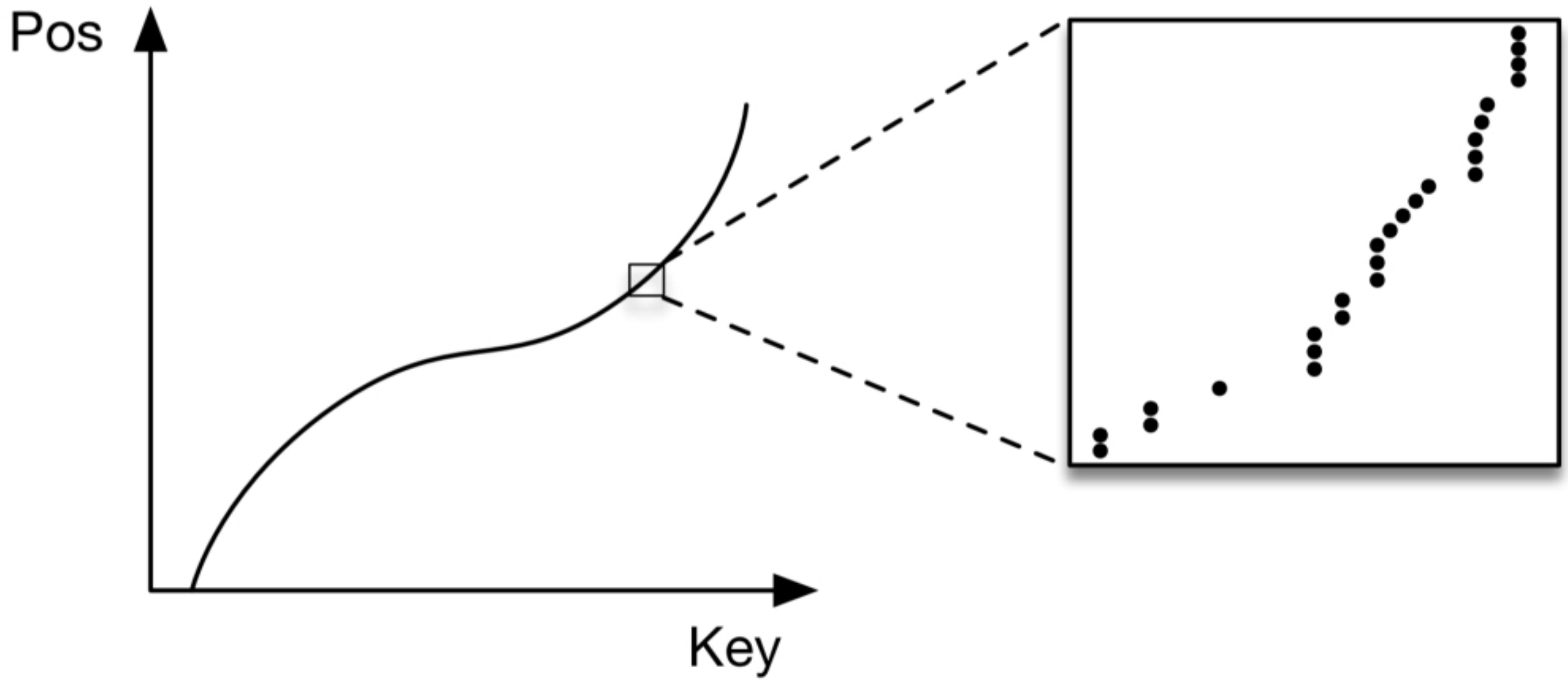
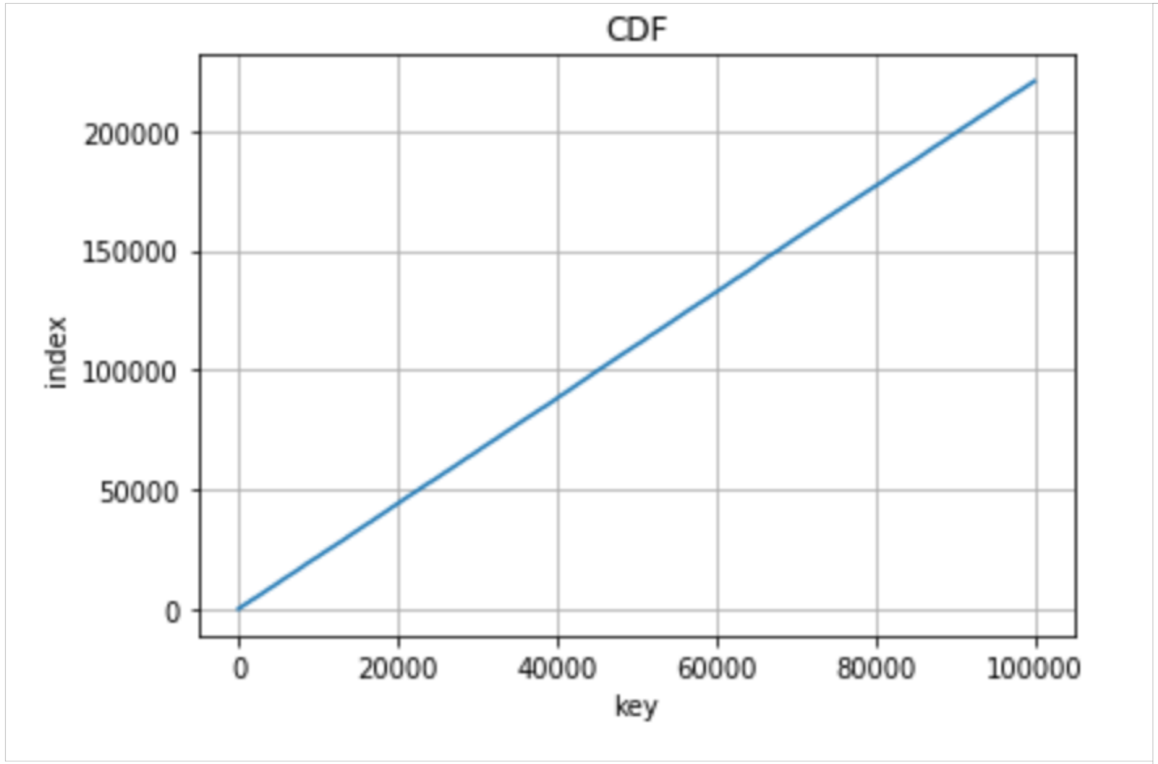
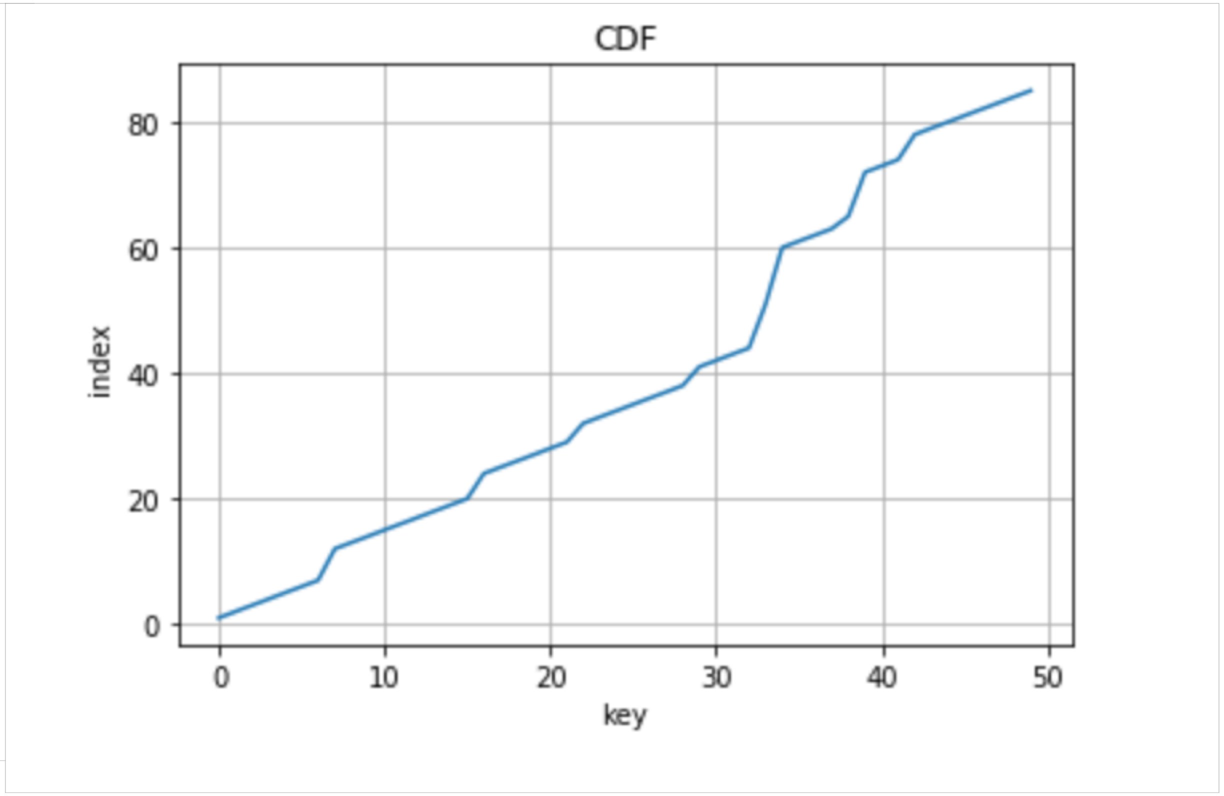


Figure 2: Indexes as CDFs

Best case vs worst case





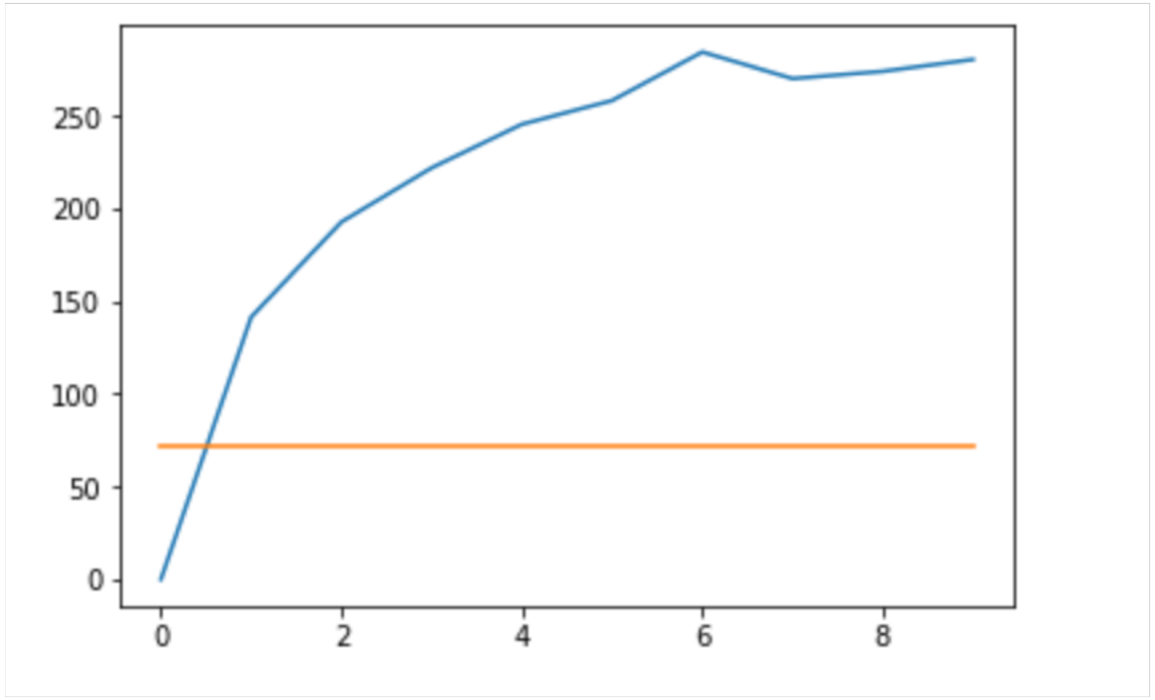
Dataset

- 100,000 keys in sorted array
- 10 levels of entropy
- averaged over 100 runs

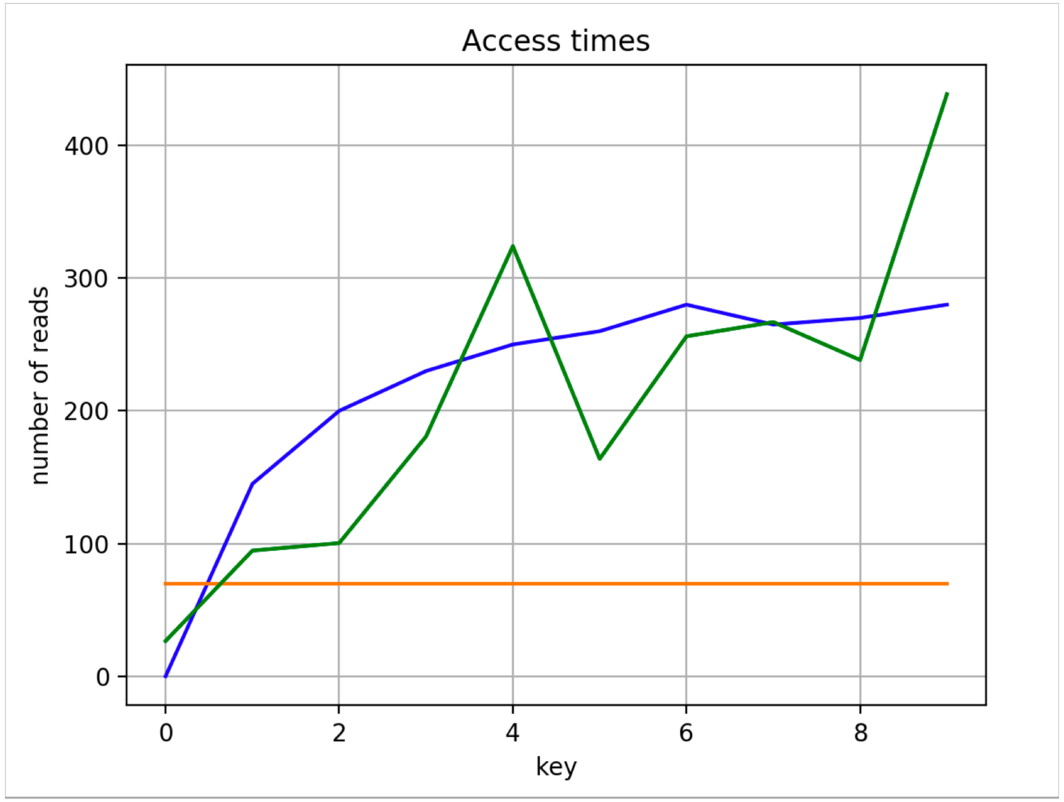
Implementation-independent evaluation

- What we really want to know: wall-clock time
- different depending on OS, hardware (SIMD), implementation of b-tree, programming language, …
- Idea: use reads as proxy
- read_time = f(n) + reads
- f(n) is the base complexity of the data structure (e.g. invocation overhead of NN of size proportional to dataset size)

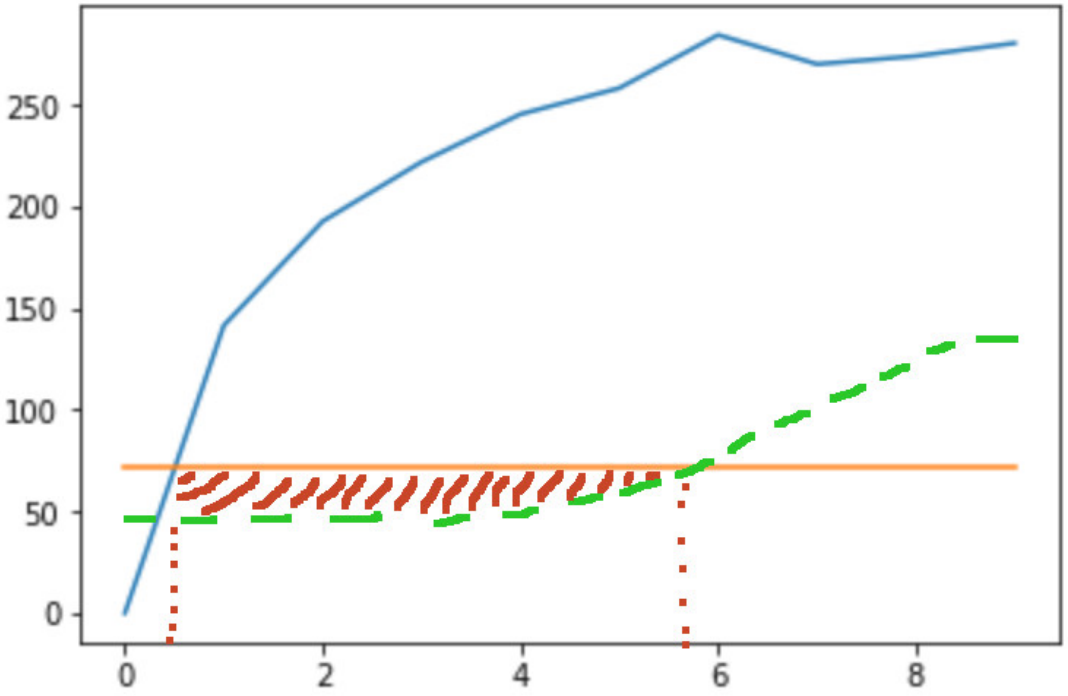
Preliminary results



Preliminary results



Hypothesis



Main references

- The paper: https://arxiv.org/abs/1712.01208
- Stanford EE380: Computer Systems Colloquium Seminar
- https://www.youtube.com/watch?v=NaqJO7rrXy0