
DATABASE

CSE3103 : Database
FALL 2020

Nazmus Sakib
Assistant Professor
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Motivation

- To have dynamic indexing structures that can evolve when records are added and deleted
  - Not the case for static indexes
  - Would have to be completely rebuilt
- Optimized for searches on ***block devices***
- Both B trees and B+ trees are not binary
  - Objective is to increase ***branching factor*** (***degree*** or fan-out) to reduce the number of device accesses

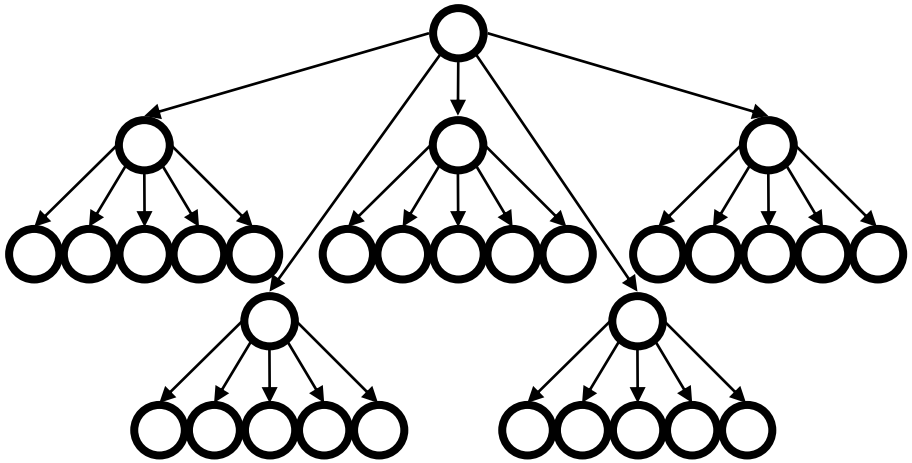
Binary vs. Higher-Order Tree

- ***Binary trees:***
  - Designed for in-memory searches.
  - Try to minimize the number of memory accesses.

- ***Higher-order trees:***
  - Designed for searching data on block devices.
  - Try to minimize the number of device accesses.
    - Searching within a block is cheap.

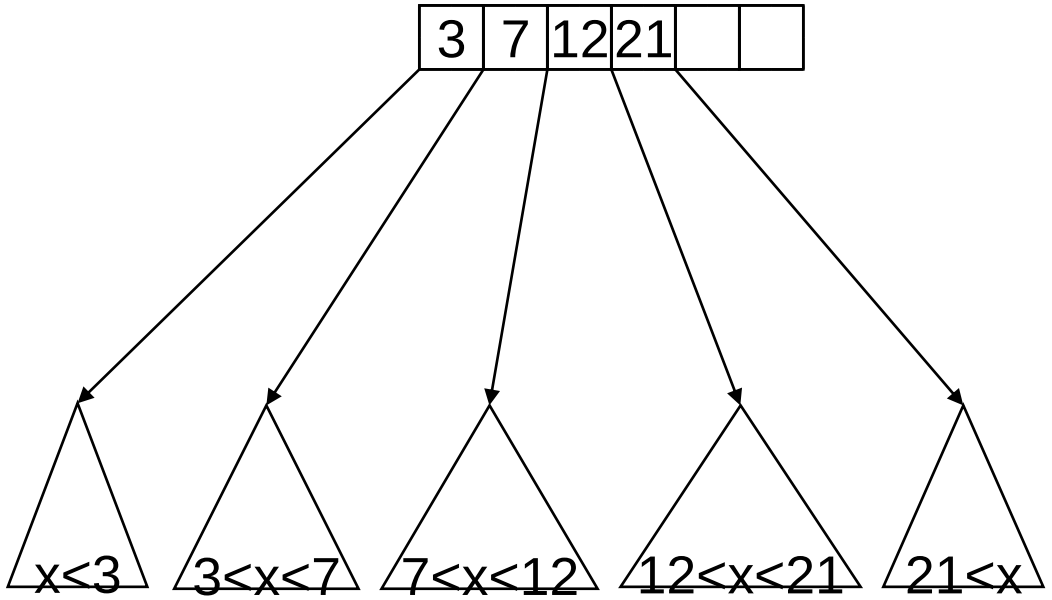
What is an M-ary Search Tree?

- ✓ Maximum branching factor of M
- ✓ Complete tree has depth = $\log_M N$
- ✓ Each internal node in a complete tree has $M - 1$ keys
- ✓ Binary search tree is a B Tree where M is 2



B Trees

- B-Trees are specialized *M*-ary search trees
- Each node has many keys
  - Sub-tree between two keys *x* and *y* contains values *v* such that $x \leq v < y$
  - binary search within a node to find correct sub-tree
- Each node takes one
  - full {*page, block, line*} of memory (disk)

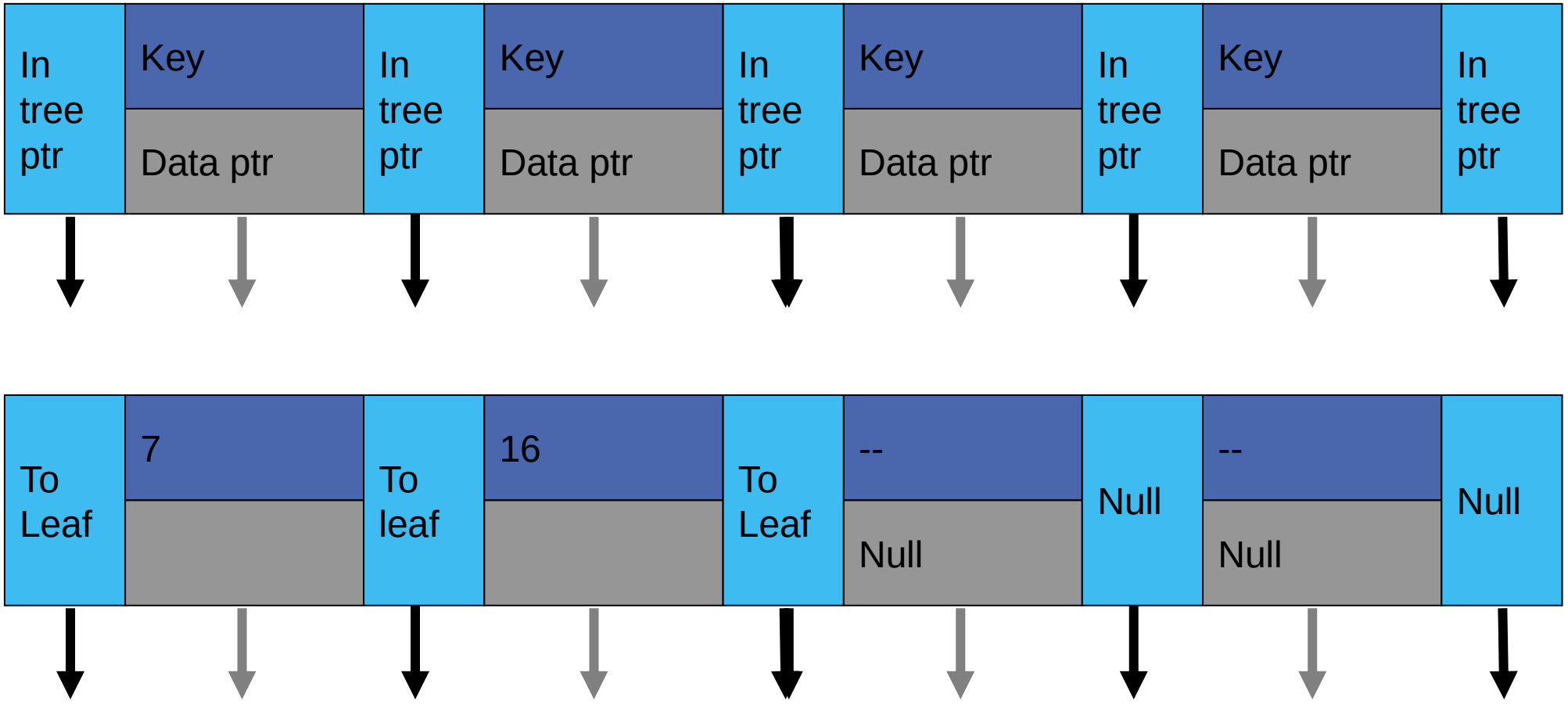


B-Tree Properties

- Properties
  - Maximum branching factor of $\boldsymbol{M}$
  - The root has between $\mathbf{2}$ and $\boldsymbol{M}$ children *or* at most $\boldsymbol{M-1}$ keys
  - All other nodes have between $\lceil \boldsymbol{M/2} \rceil$ and $\boldsymbol{M}$ record
  - Keys + Data
- Result
  - tree is $\mathbf{O(\log M)}$ deep
  - all operations run in $\mathbf{O(\log M)}$ time
  - operations pull in about $\boldsymbol{M}$ items at a time

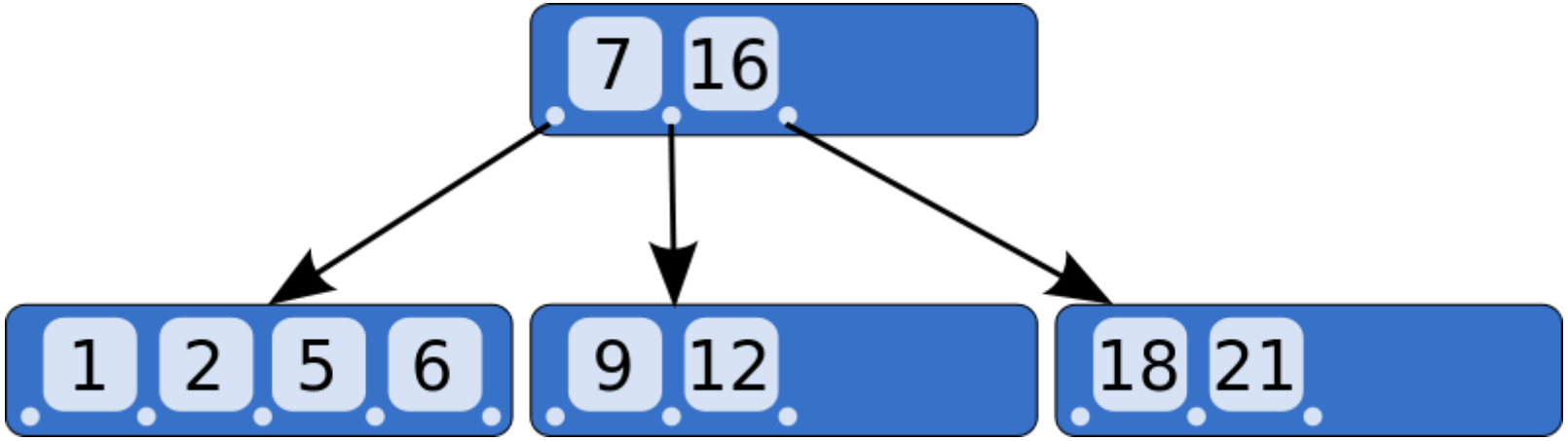
©NAZMUS SAKIB

In Reality



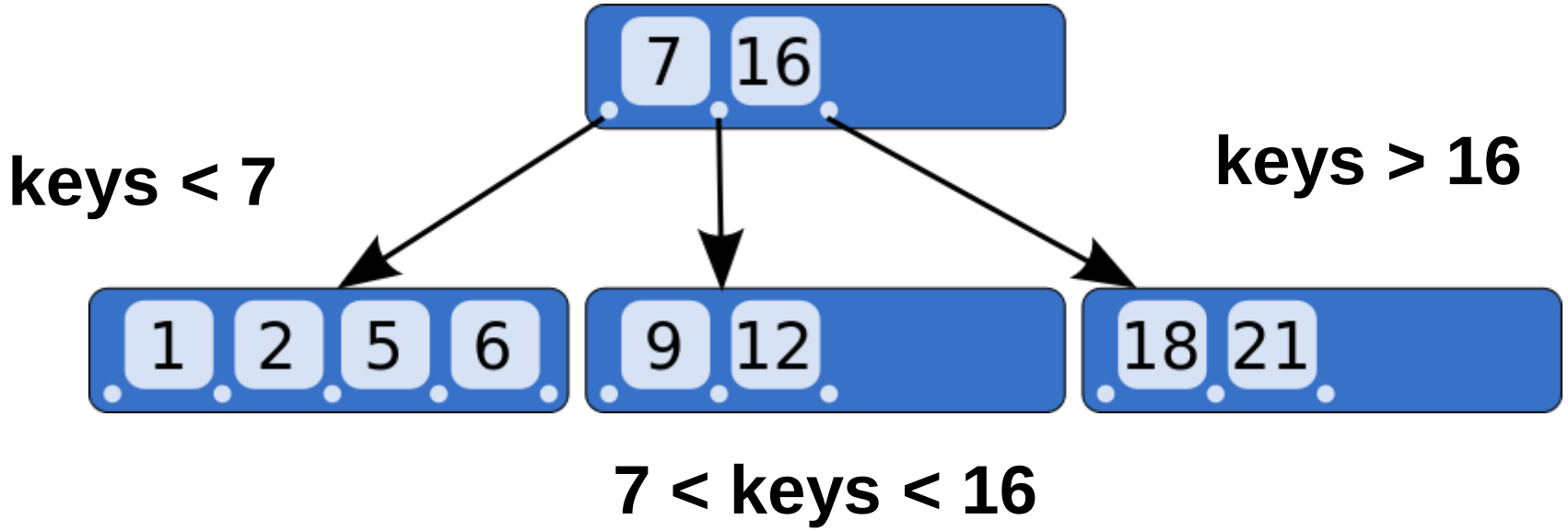
©NAZMUS SAKIB

A very small B tree



Bottom nodes are leaf nodes: all their pointers are NULL

Searching the tree




DBMS
data
distributed
parallel
database
shared
system
Distributed
architecture
DBMSs
network
systems
processing
performance
processors
access
centralized
multiple
sites
Parallel
processor
may
major
typically
fragments
fact
across
require
physically
within
users
site
machines
like
Database
applications
computers
control
scalability
approach
improve
distinction
logical
nothing
DDBMS
Shared
memory
databases
development
definition
nodes
application
management
transparent
also
many
replicated
make
sometimes
computing
single
one
split
order
local
provide
often
Management
number
dis
user
makes
bottleneck
interconnection
Figure
administered
mode
organizational
consists
logically
processors
decentralized
directly
com
Date
operational
referred
disks
large
islands
resources
remote
defined
controlled
based
structure
collection
provides
global
associated
stored
disk
computer
partitioned
used
least
must
System
technology
capable
incompatible
way
per
eliminates
reliability
distribution
communication
scalable
architectures
discuss
share