
DATABASE

CSE3103 : Database

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

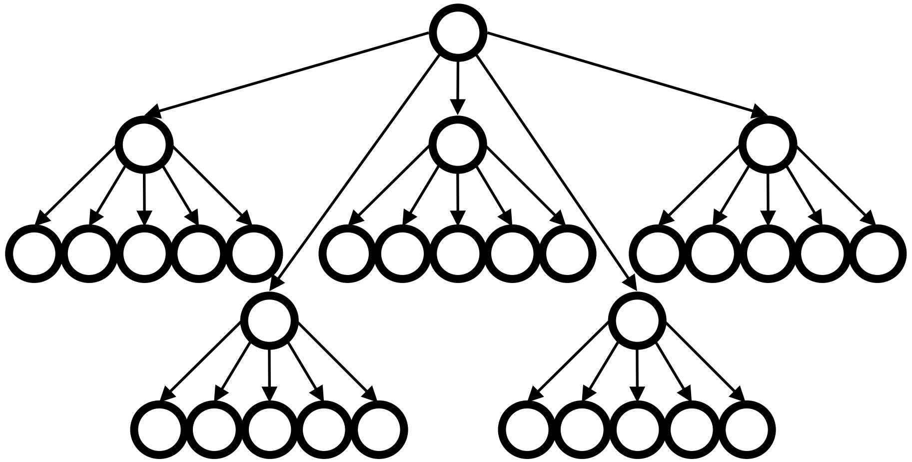
Binary vs. Higher-Order Tree

- ***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.

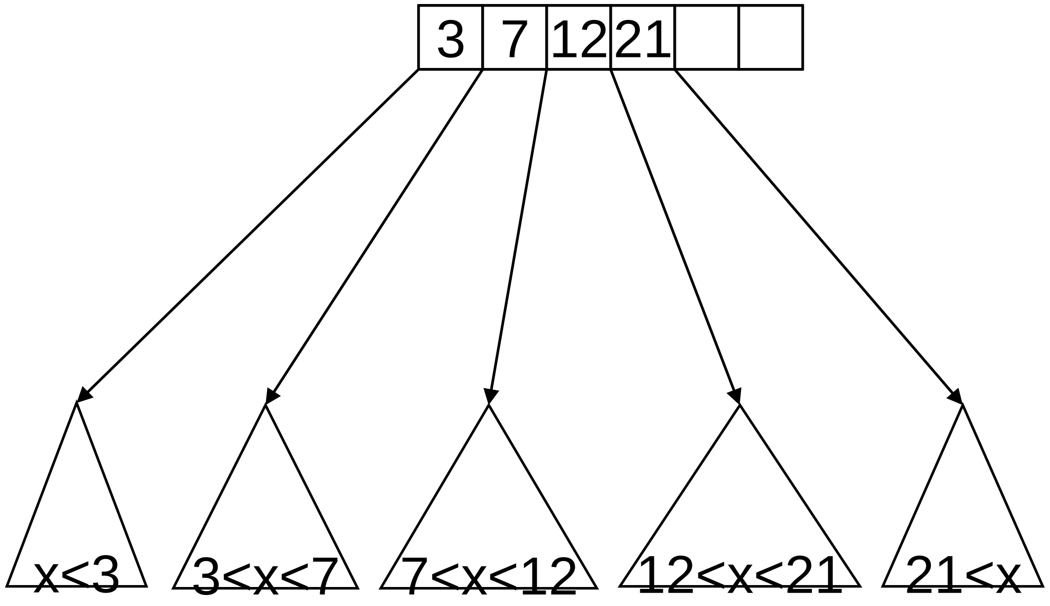
What is an M-ary Search Tree?

- ✓ 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

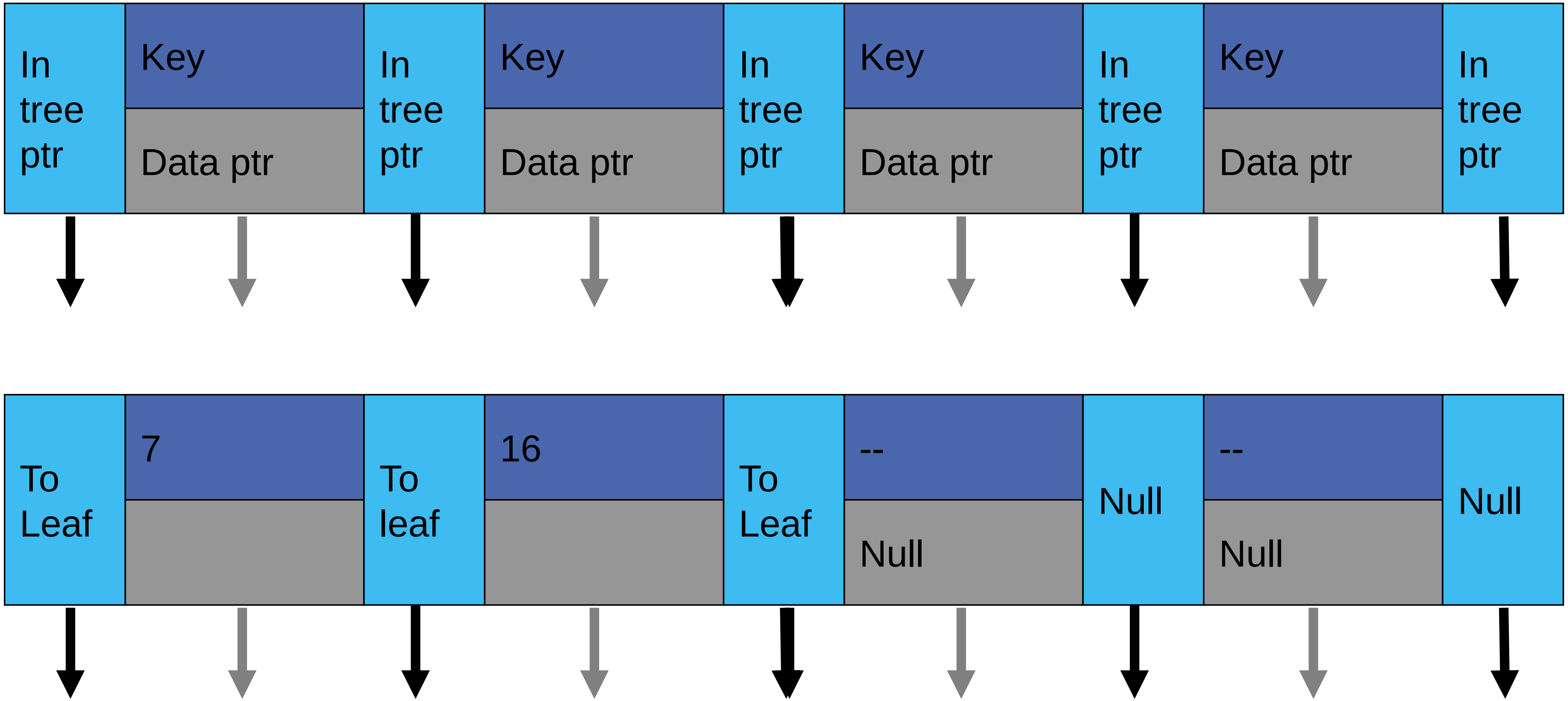
- B-Trees are specialized *M*-ary search trees
- Each node has many keys
  - Sub-tree between two keys x and y contains values v such that $x \le v < y$
  - binary search within a node to find correct sub-tree
- Each node takes one
  - full {*page, block, line*} of memory (disk)



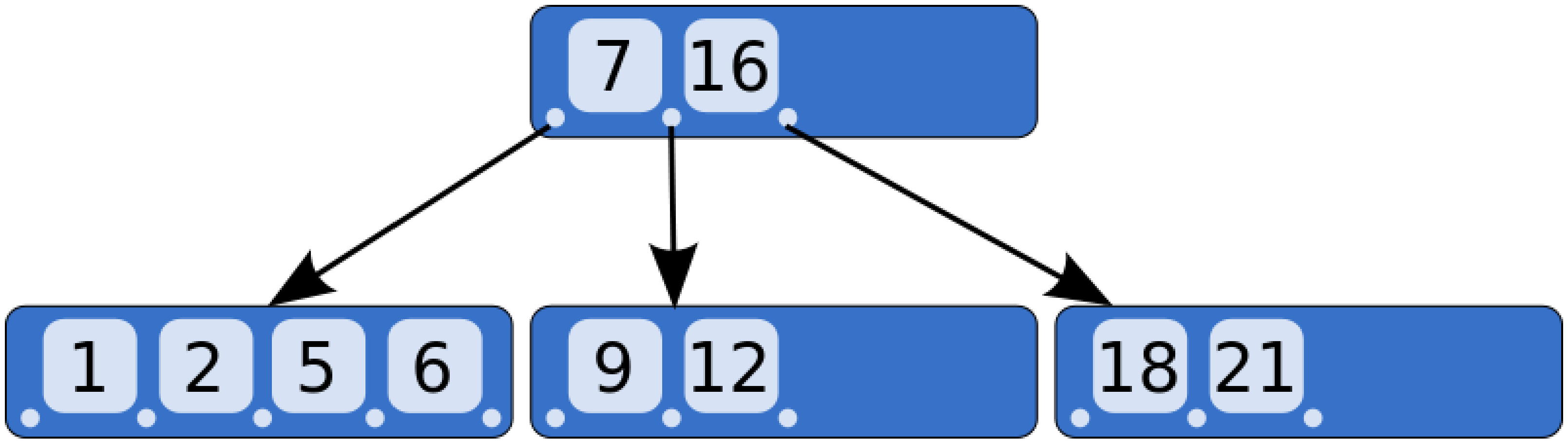
B-Tree Properties

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

In Reality

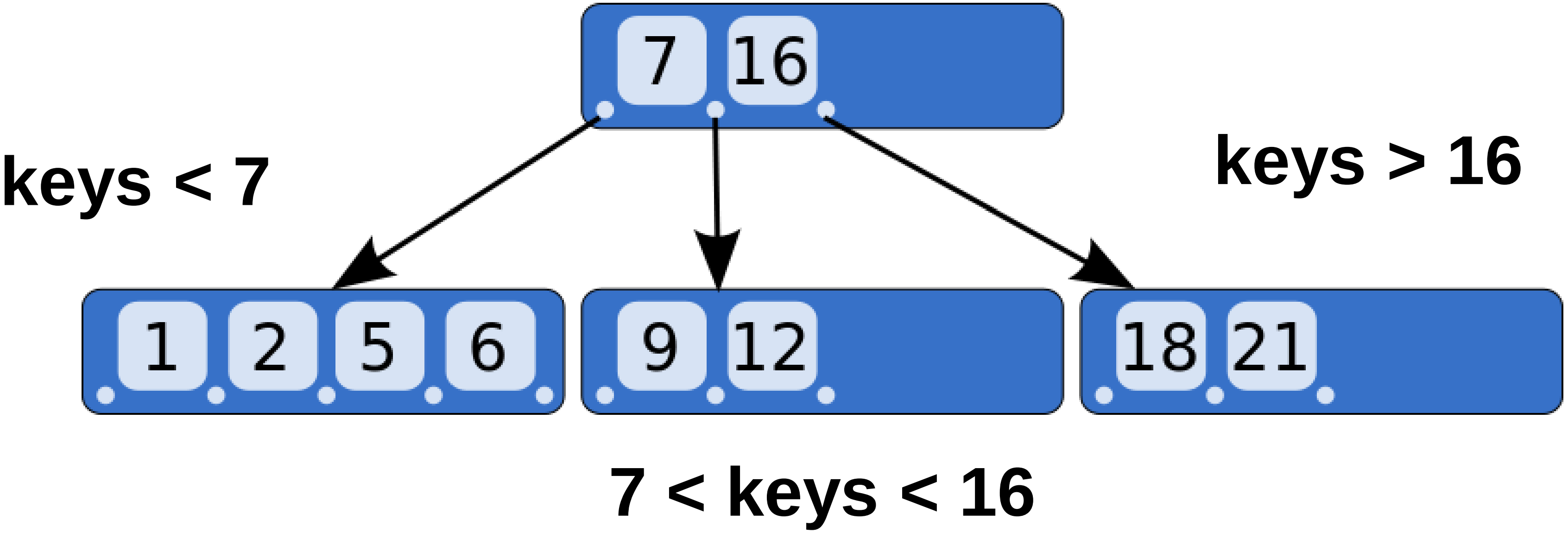

In tree ptr
Key
Data ptr
In tree ptr
Key
Data ptr
In tree ptr
Key
Data ptr
In tree ptr
Key
Data ptr
In tree ptr
To Leaf
7
To leaf
16
To Leaf
--
Null
Null
--
Null
Null

A very small B tree

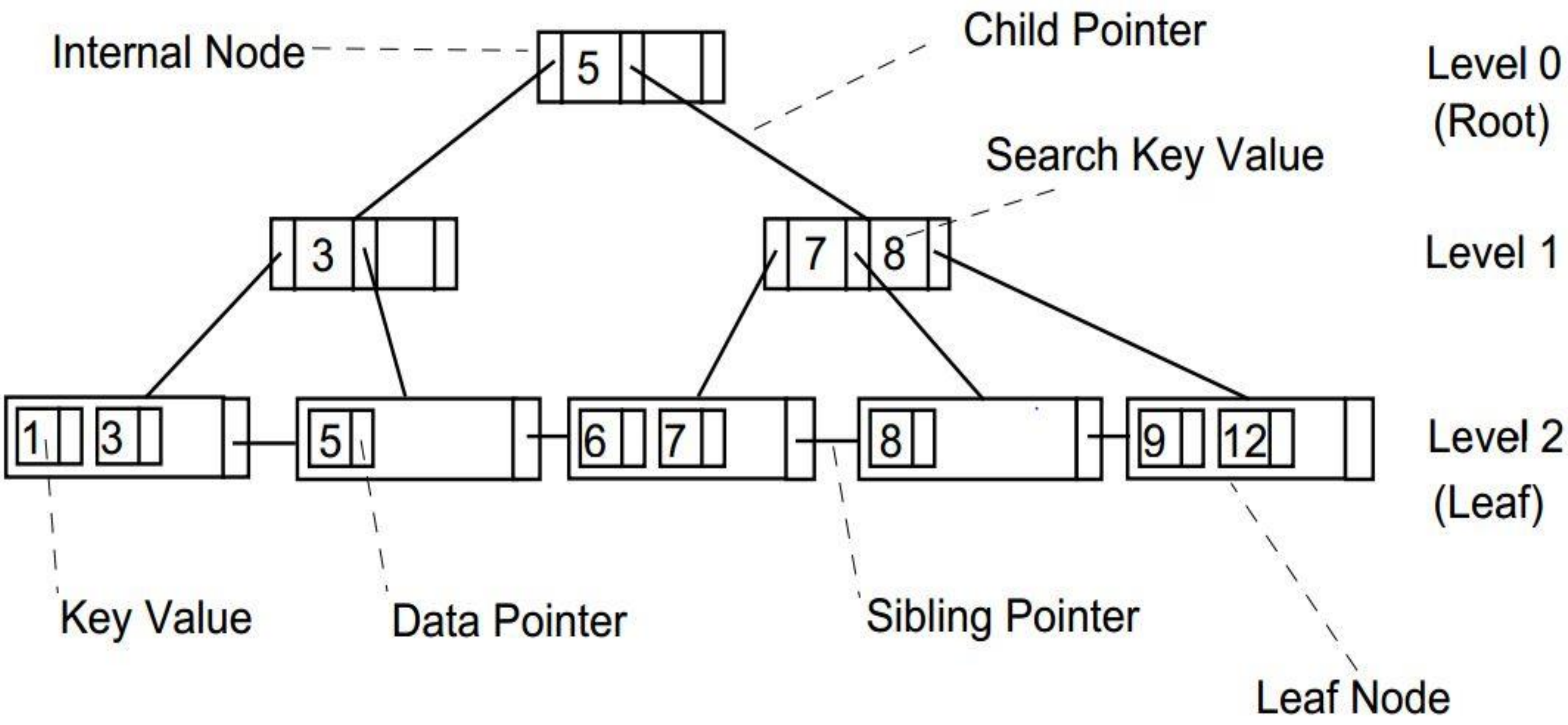


Bottom nodes are leaf nodes: all their pointers are NULL

Searching the tree



B+ Tree



Internal Node
Child Pointer
Level 0
(Root)
Search Key Value
Level 1
Level 2
(Leaf)
Key Value
Data Pointer
Sibling Pointer
Leaf Node
5
3
7
8
1
3
5
6
7
8
9
12


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