CS 186 Discussion 4

Files and Indexes (B+ Trees)

Logistics

- Vitamin 3 due tonight (2/16) 11:59 PM
- Homework 3 due next Thursday
  - Start *soon*
- Midterm 1 in two weeks

Heap Files

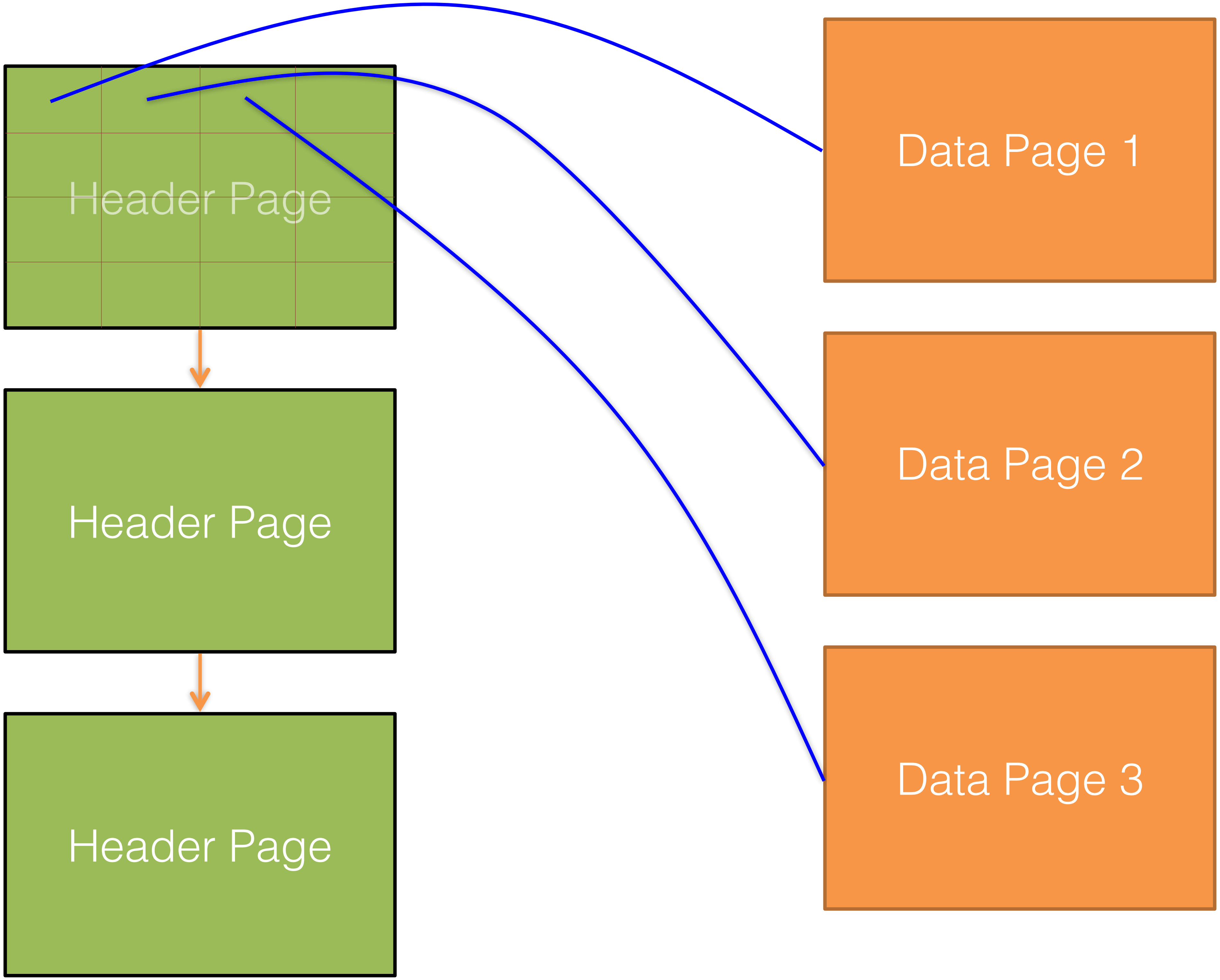
Unordered Files

Heap Files

- Within a heap file, keep track of pages
- Within a page, keep track of records
  - Also keep track of free space

- *RID (Record ID) = <page id, slot #>*

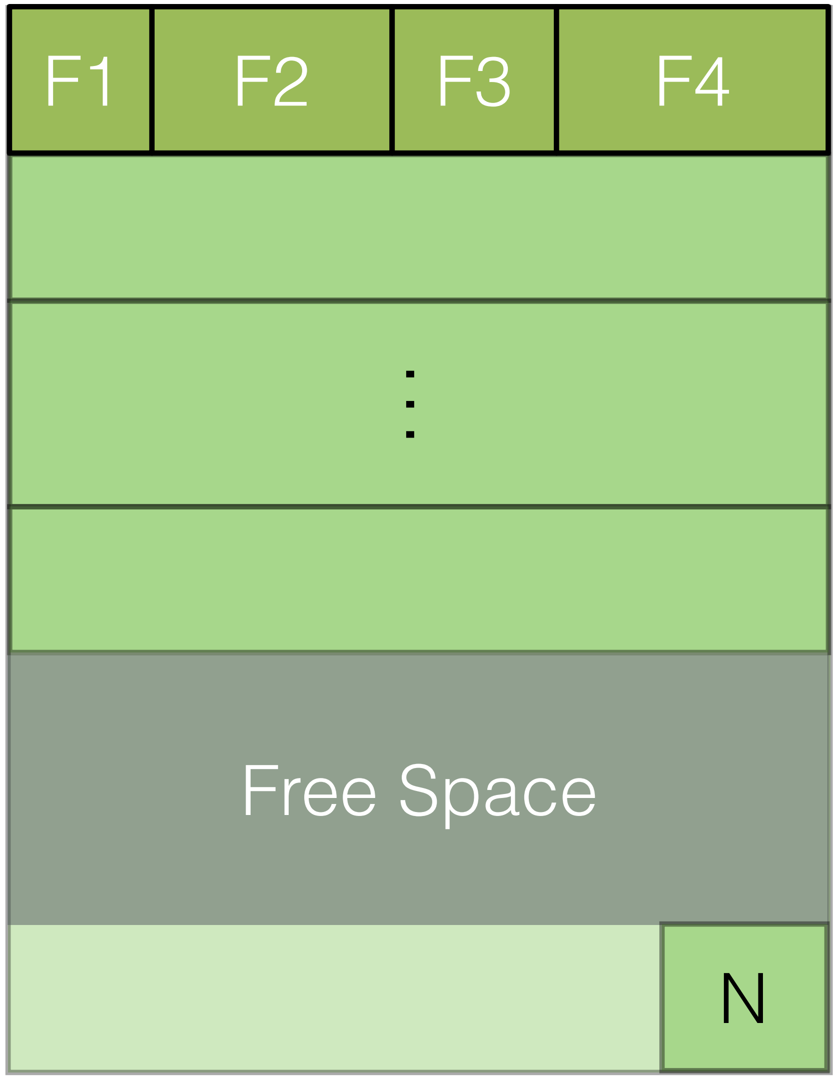
Page Directory



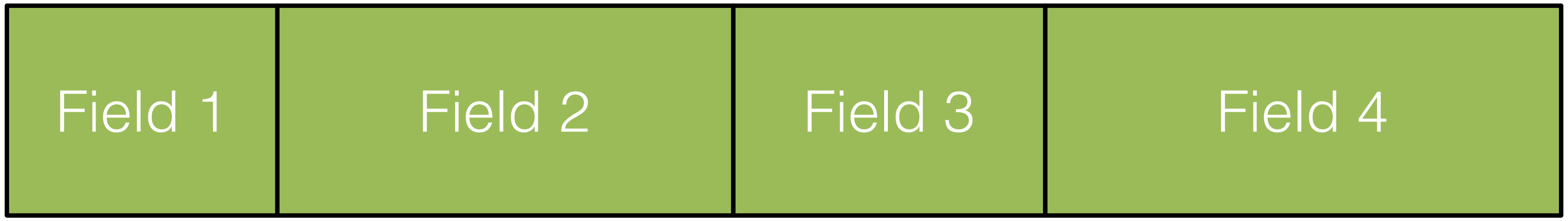
Keep # free bytes on page in directory entries

Fixed Length Records

- Fixed record length
- Consistent field length

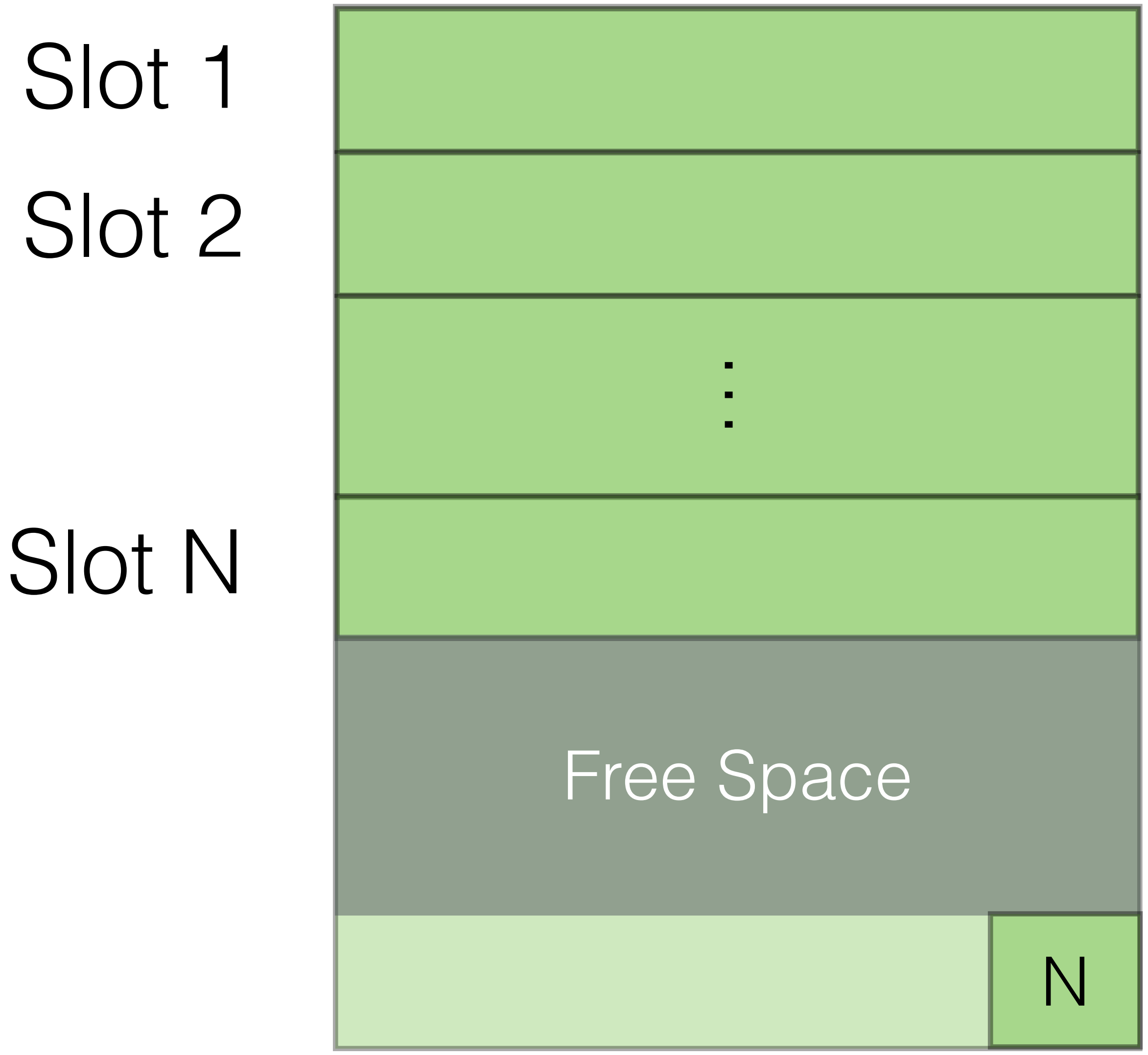


Keep field type info
in *system catalog*

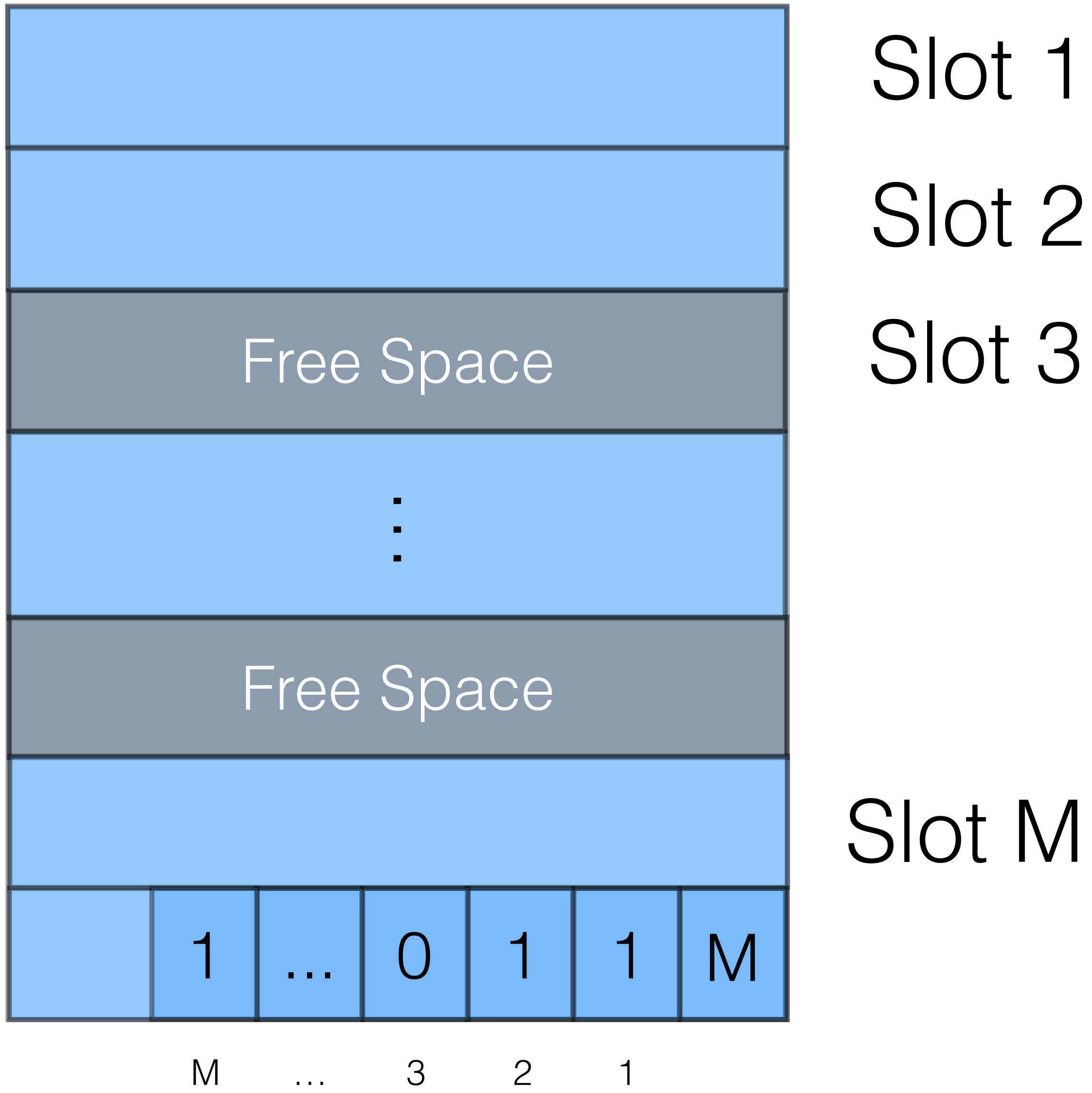


<- Record

Page Format w/ Fixed-Length Records



Packed



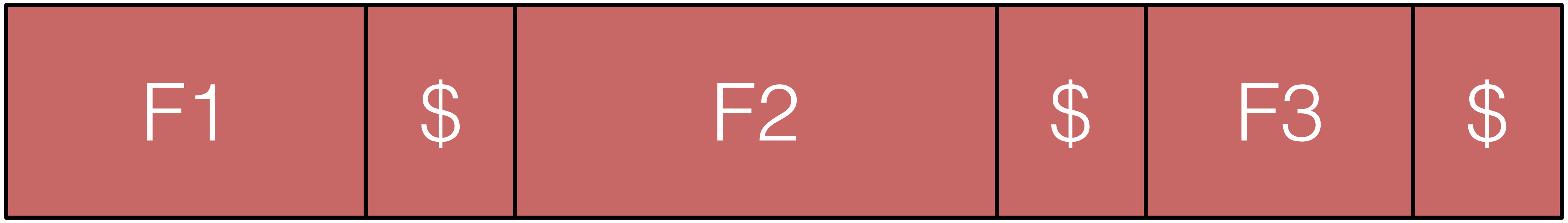
Unpacked (Bitmap)

Diagrams by Michelle Nguyen

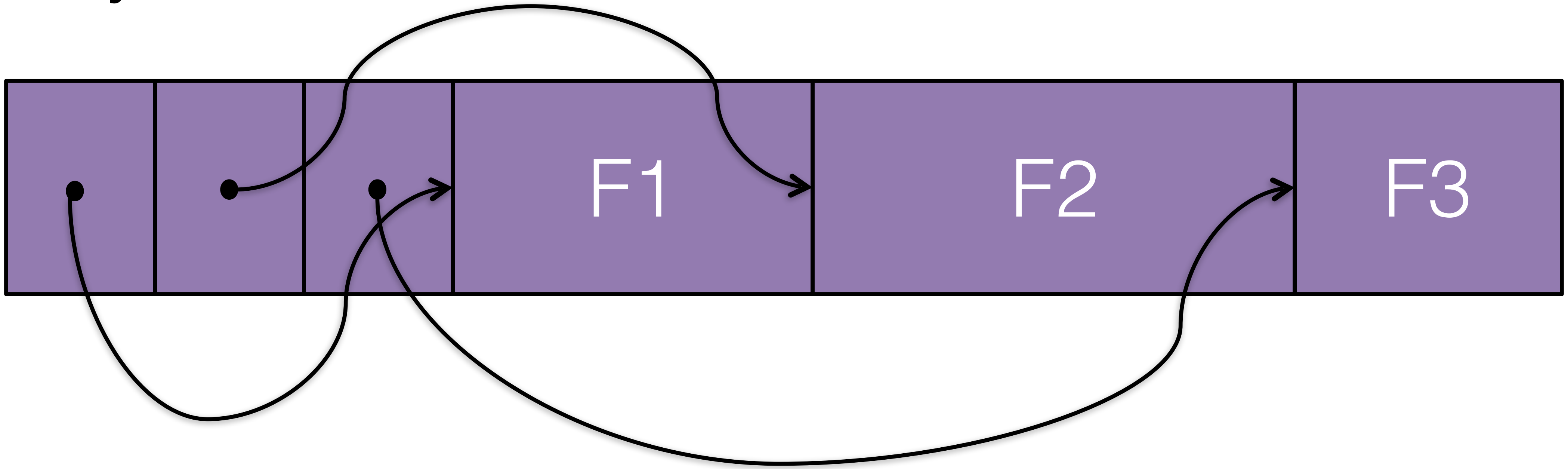
Headers can be at the top of the page too!

Variable-Length Records

- Delimit fields with special characters

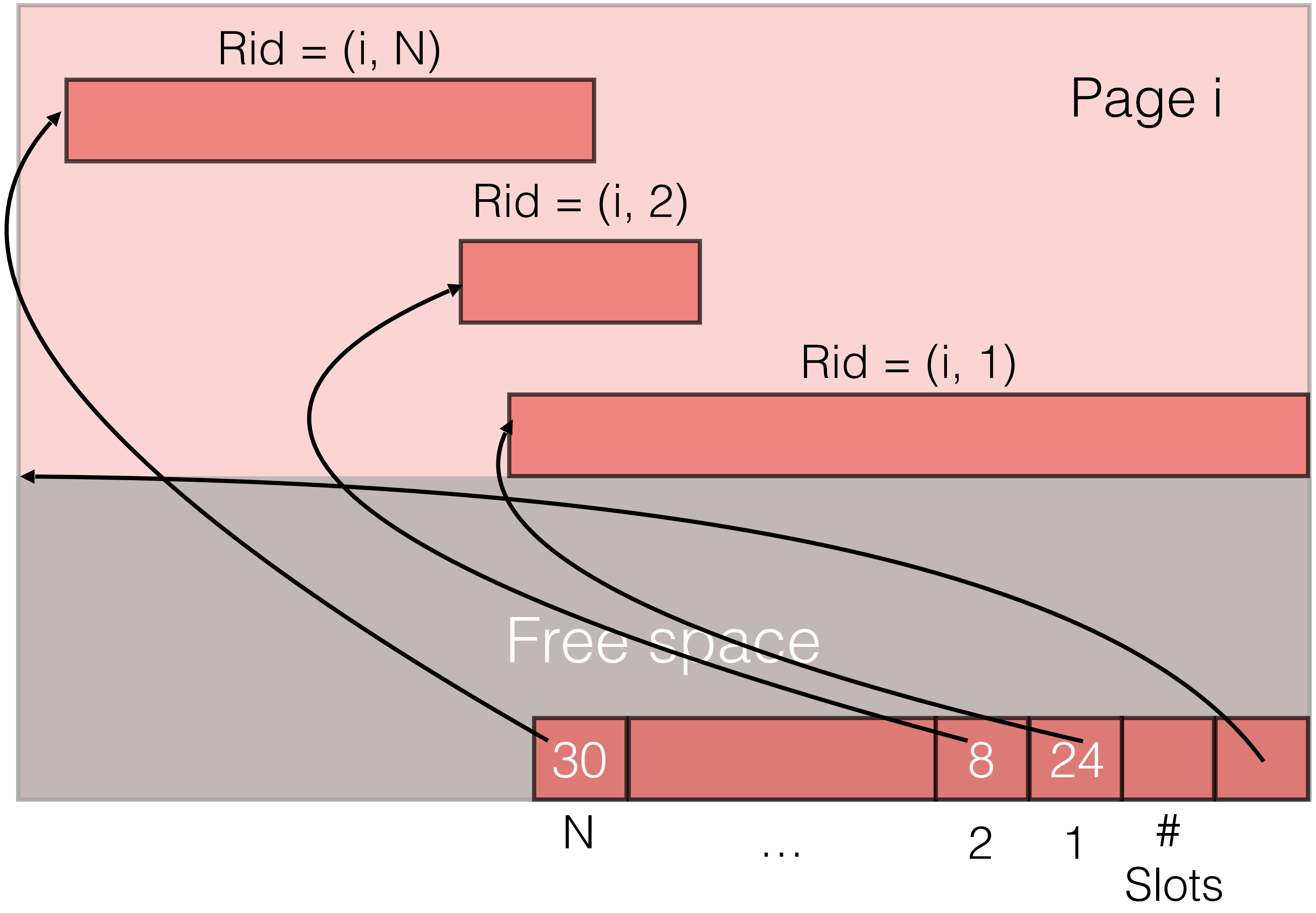


- Array of field offsets



  - Typically preferred choice

Page Format w/ Variable-Length Records



Want to be able to move records around!

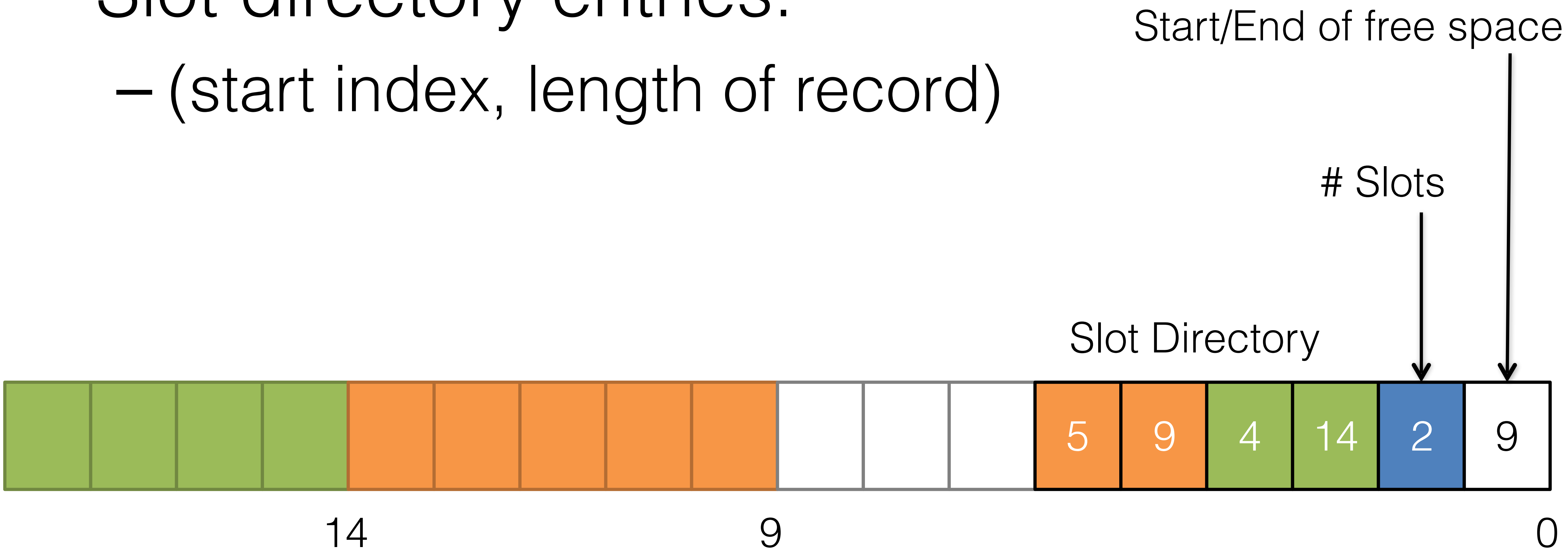
Pointer to start of free space

Slotted Page

Diagram by Michelle Nguyen

Slotted Page – Detailed View

- Slot directory entries:
  - (start index, length of record)

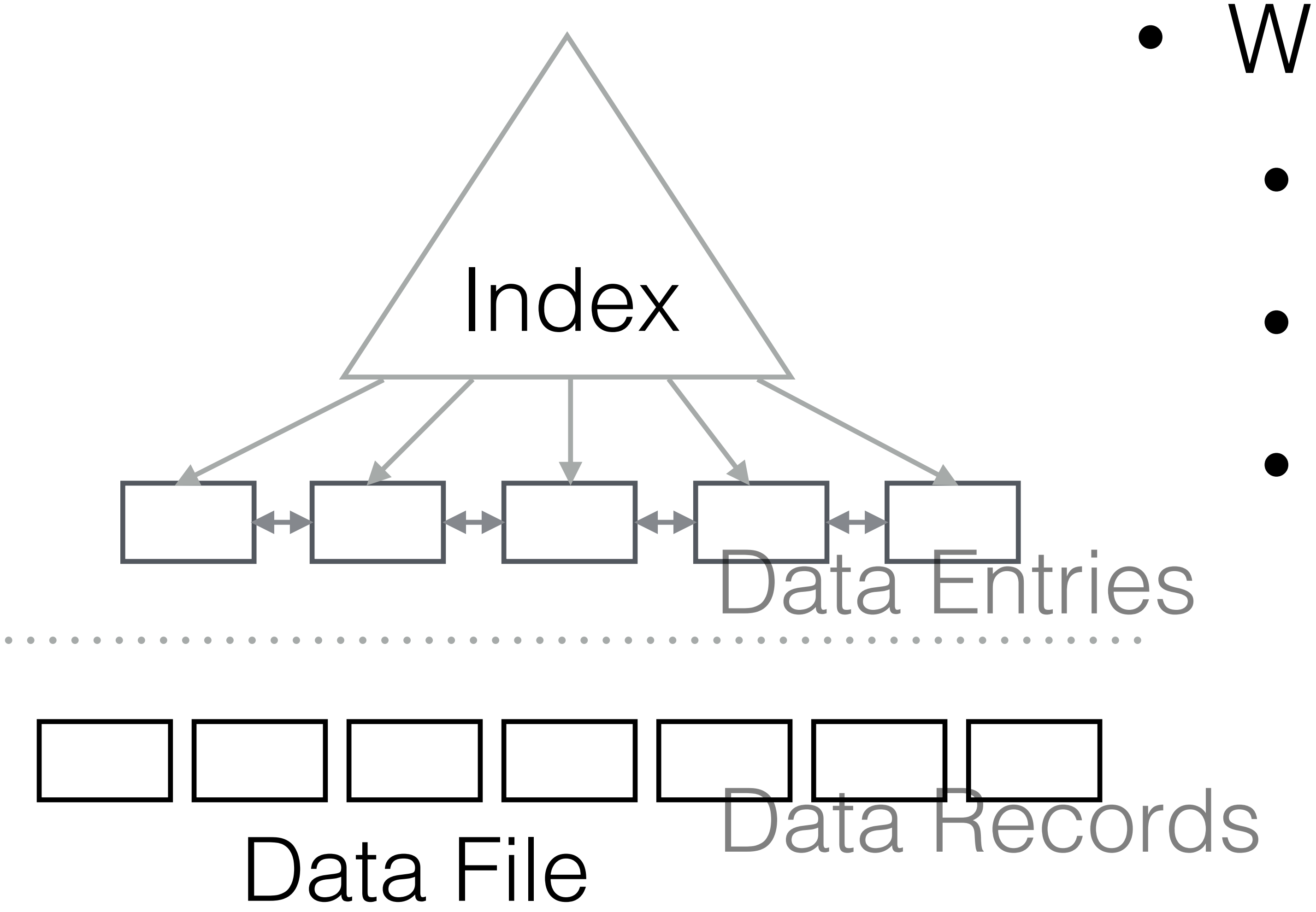


Cost of Operations

	Heap File	Sorted File	Clustered Index
Scan all records	B	B	(3/2) B = 1.5B
Equality Search	0.5 B	$\log_2 B$	$\log_F 1.5B + 1$
Range Search	B	$\log_2 B$ + #match pg	$\log_F 1.5B$ + #match pg
Insert	2	$\log_2 B + B$	$\log_F 1.5B + 2$
Delete	0.5B + 1	$\log_2 B + B$	$\log_F 1.5B + 2$

Indexes

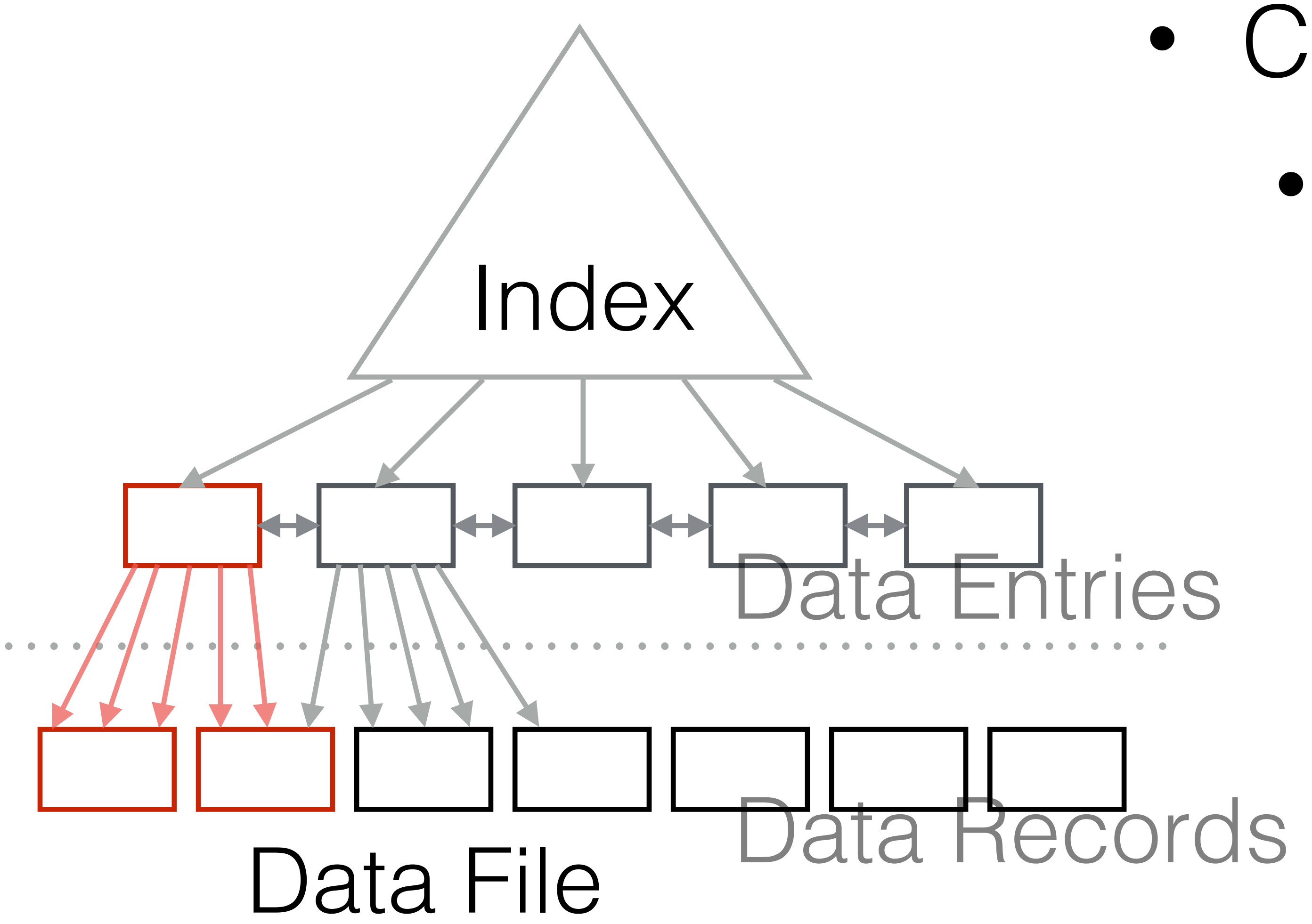
Index Organization



- What goes in data entries?
  - Alt. 1: Actual records
  - Alt. 2: <key, rid>
  - Alt. 3: <key, [rids]>

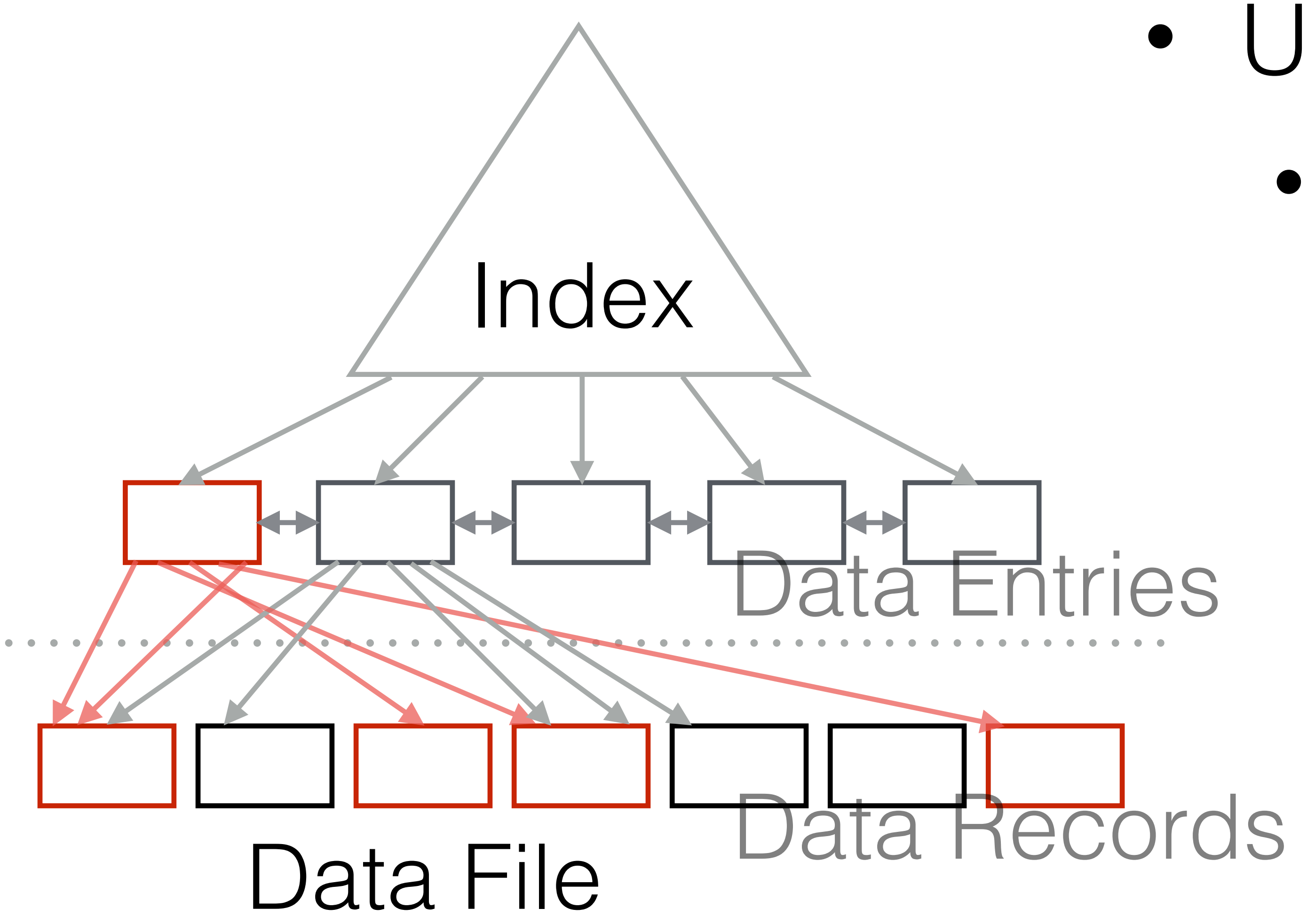
- What's sorted by key?

Index Organization



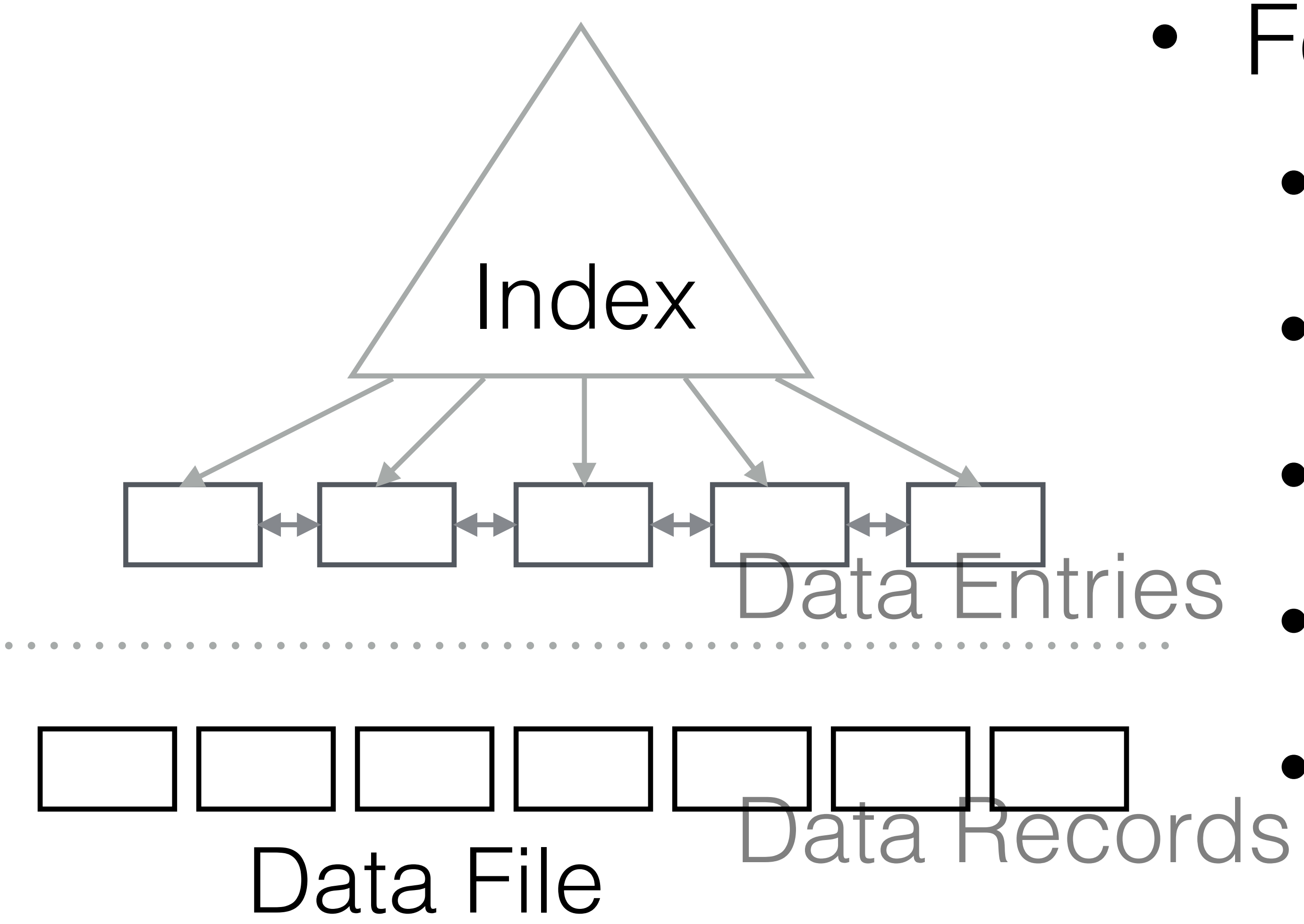
- Clustered:
  - Data records also (approx.) sorted by key

Index Organization



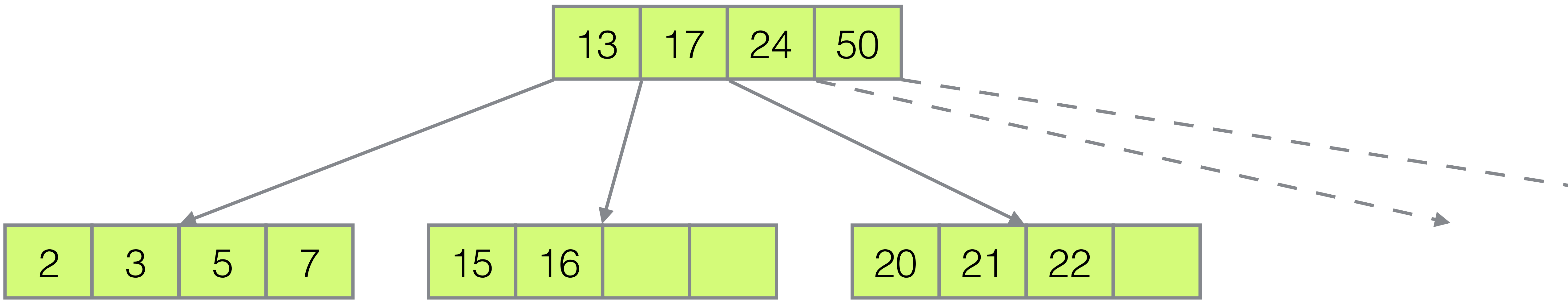
- Unclustered:
  - Data records are in any order, not sorted by key
- Why bother?

Index Organization



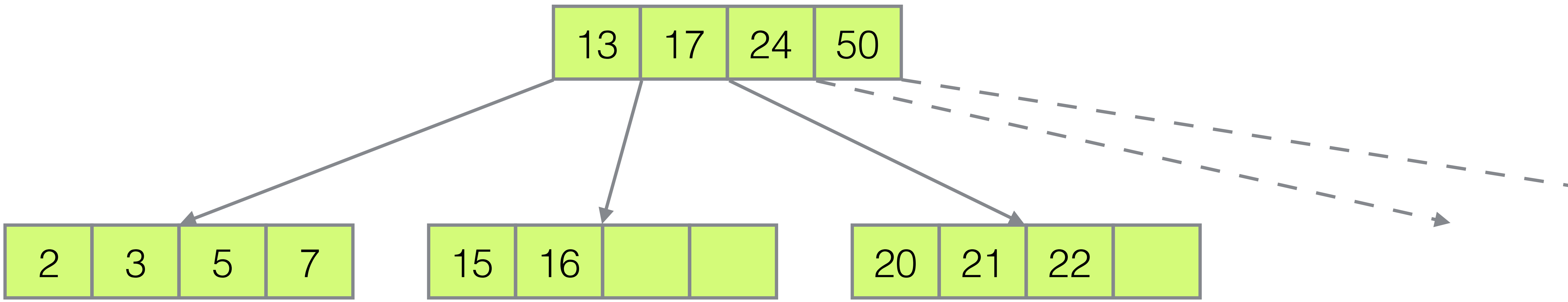
- For each data file:
  - How many clustered?
  - How many unclustered?
  - How many Alt. 1?
  - How many Alt. 2?
  - How many Alt. 3?

B+ Trees



- Fanout [F] =
- Order [d] =
- Height =
- IOs to retrieve data record =

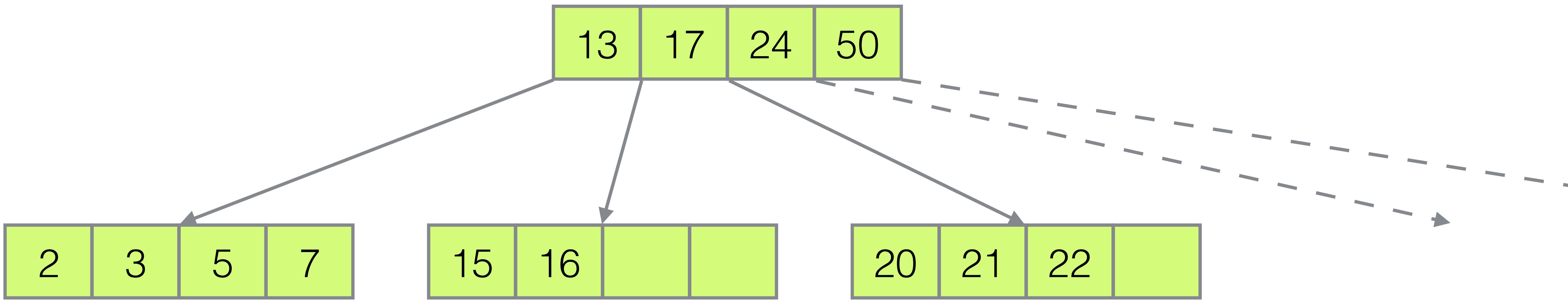
B+ Trees



- Fanout [F] = 5
- Order [d] = 2 —> each node filled to [d, 2d]
- Height = 1*
- IOs to retrieve data record = height + 1 = 2

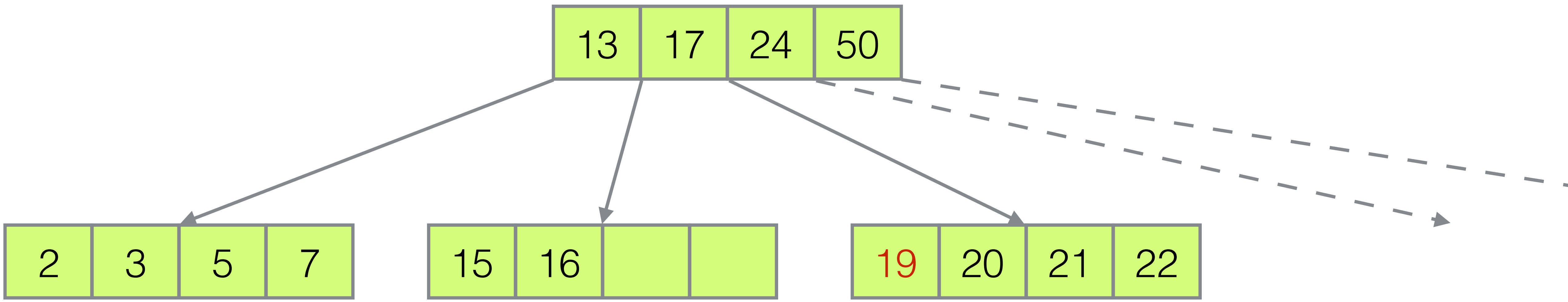
B+ Trees

Insert 19



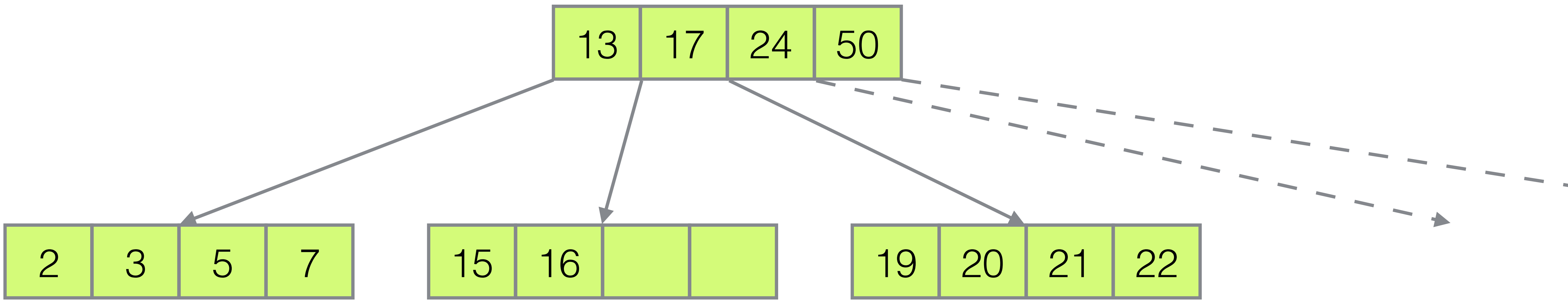
B+ Trees

Insert 19



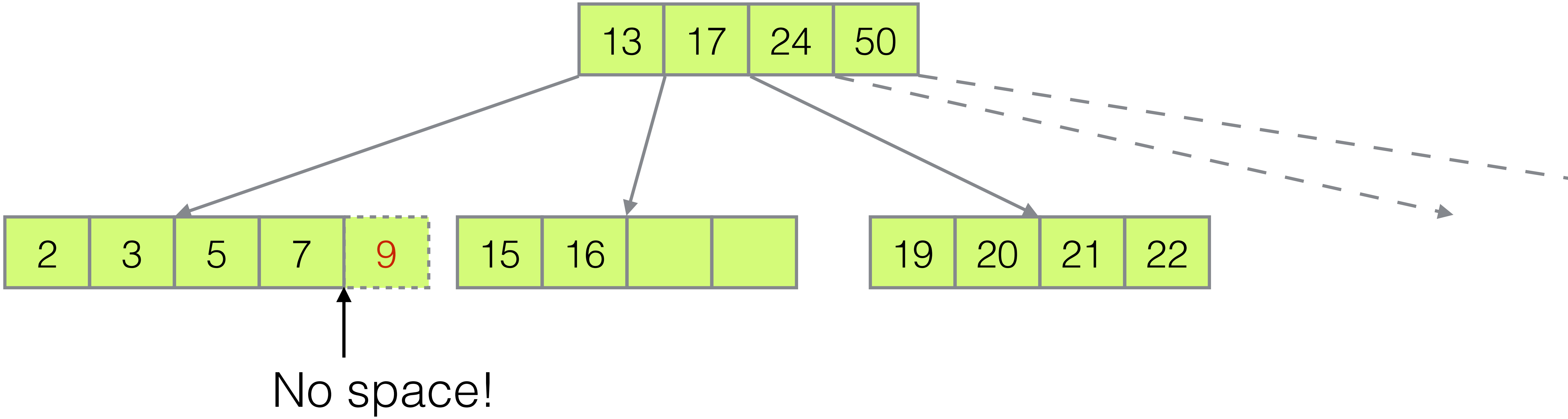
B+ Trees

Insert 9



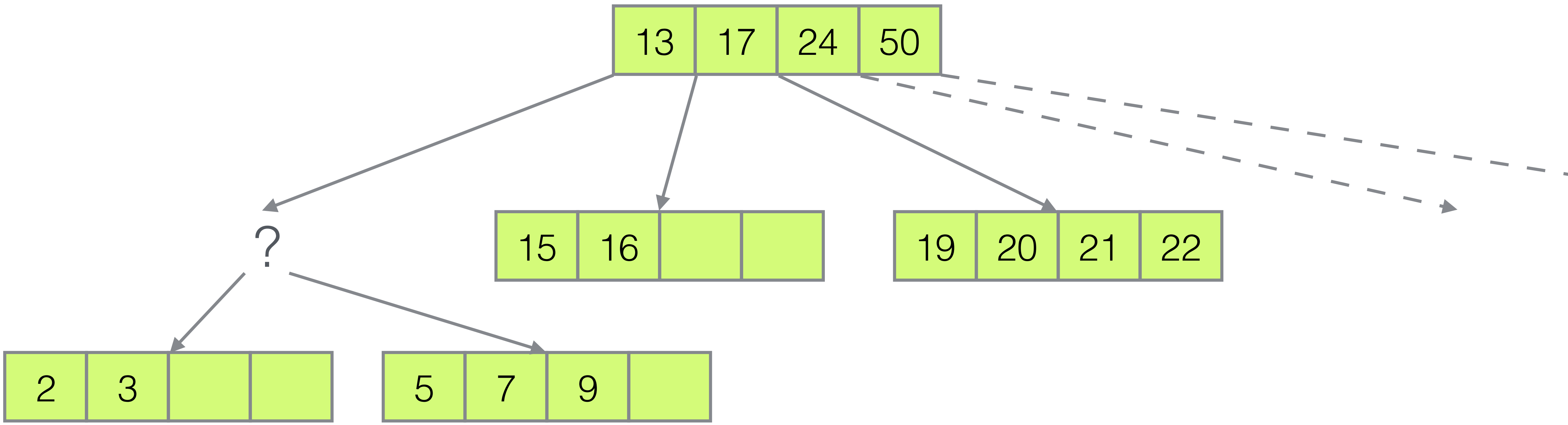
B+ Trees

Insert 9



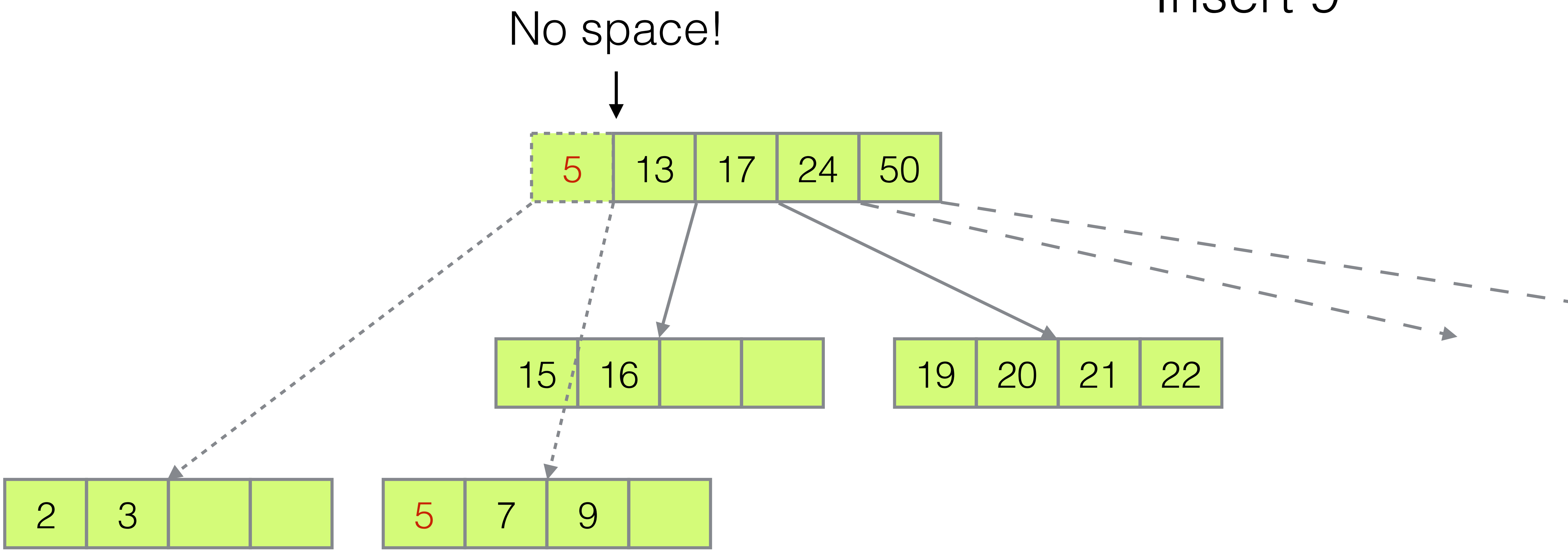
B+ Trees

Insert 9



B+ Trees

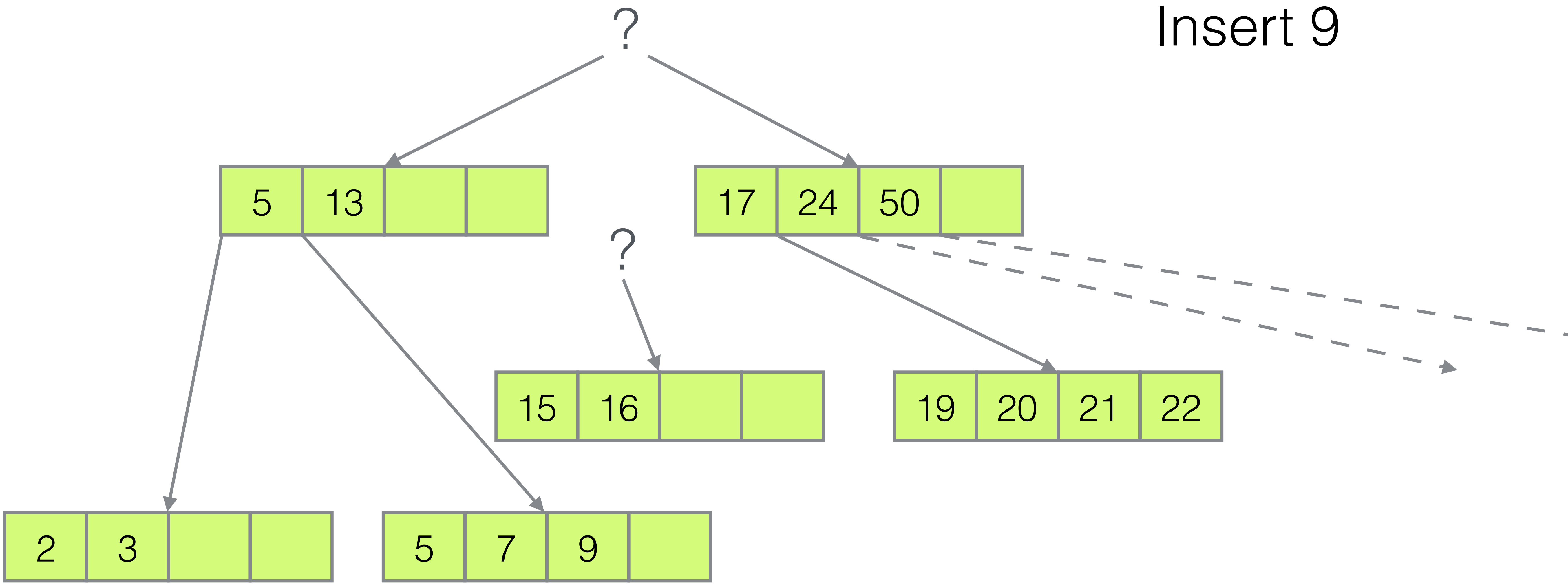
Insert 9



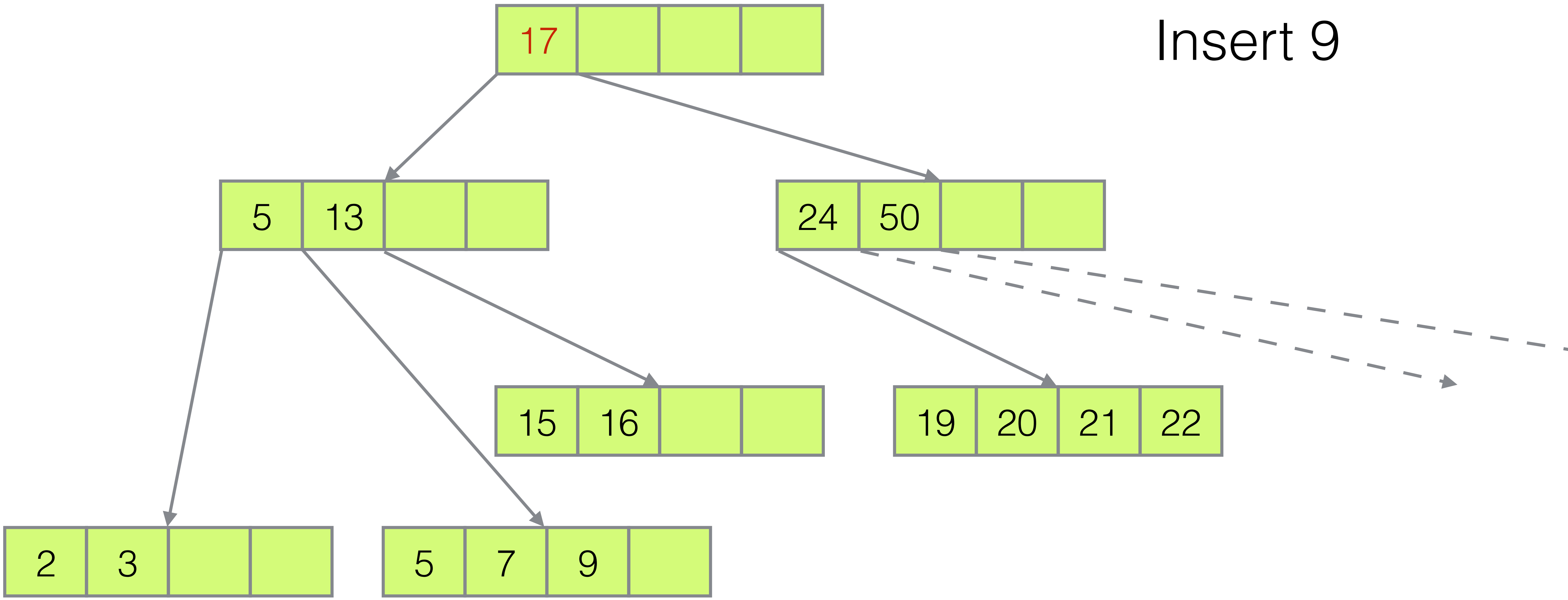
Leaf: *Copy* middle key up

B+ Trees

Insert 9



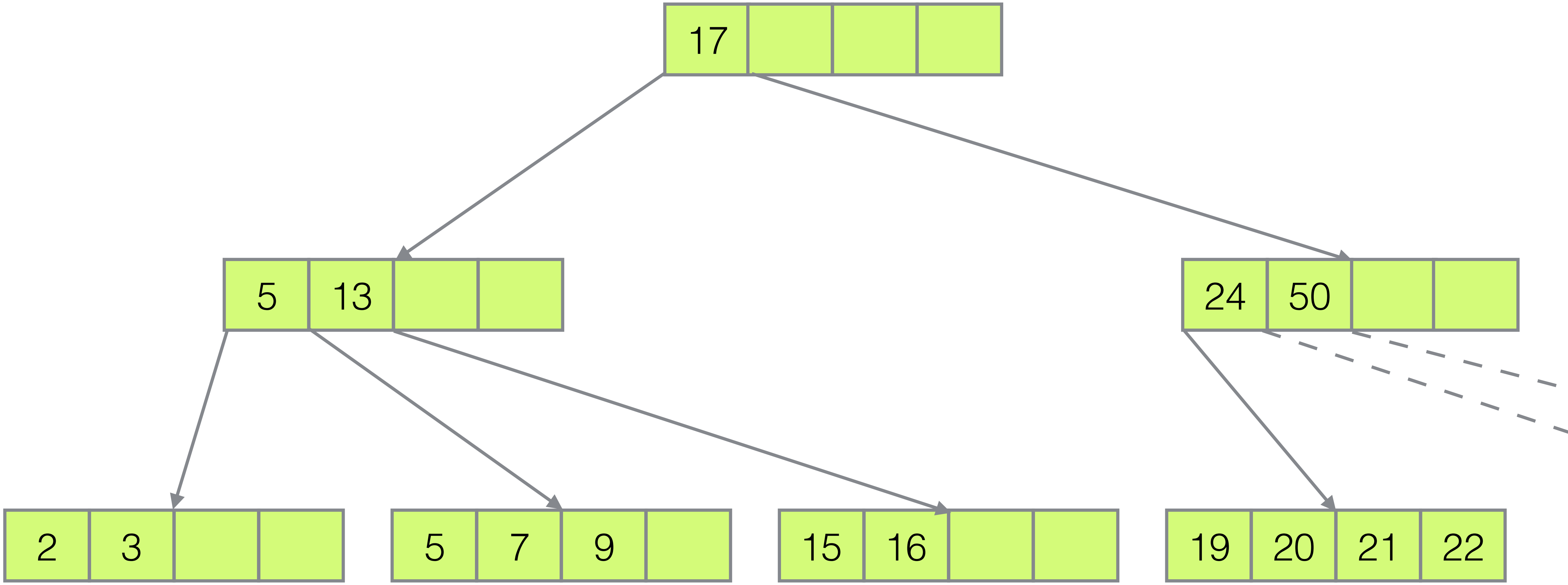
B+ Trees



Insert 9

Non-leaf: *Push* middle key up

B+ Trees



What have we maintained?