B Tree Analysis

CS 225 - Fall 2025

Mattox Beckman / Based on slides from Brad Solomon

Objectives

- Explain the importance of m in the B Tree
- Explain how delete preserves B Tree properties
- Analyze the performance of a B Tree

Definitions

A B Tree of order m is an m-ary tree

- All keys are ordered
- A node contains no more than $m - 1$ keys
- Internal nodes have exactly **one more child than keys**
  - Intuition: how many keys / childern in a BST?
- All leaves are on the same level

Find (Review)

Base Case

- root empty $\Rightarrow$ return
- leaf $\Rightarrow$ do array find
-

Recursive Case

- Array find for first $\geq$ key.
- Recurse to appropriate child
- Child index and key index are the same!

Find (Review)

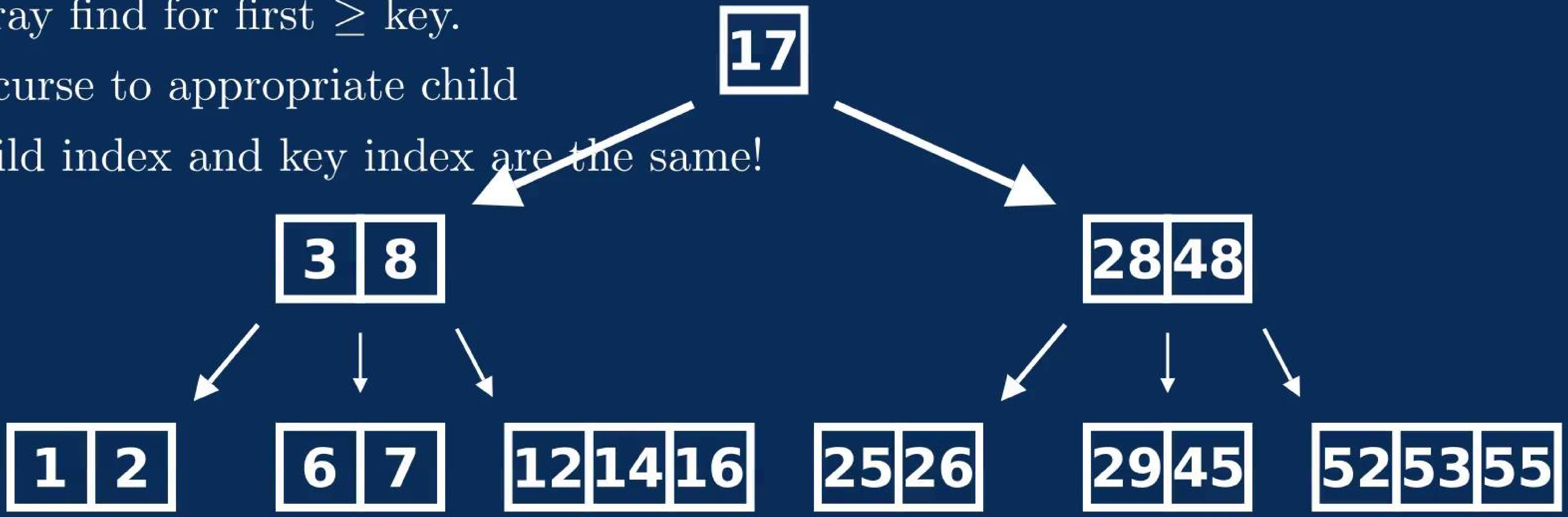
Base Case

- root empty $\Rightarrow$ return
- leaf $\Rightarrow$ do array find
-

Recursive Case

- Array find for first $\geq$ key.
- Recurse to appropriate child
- Child index and key index are the same!

find(6)



Find (Review)

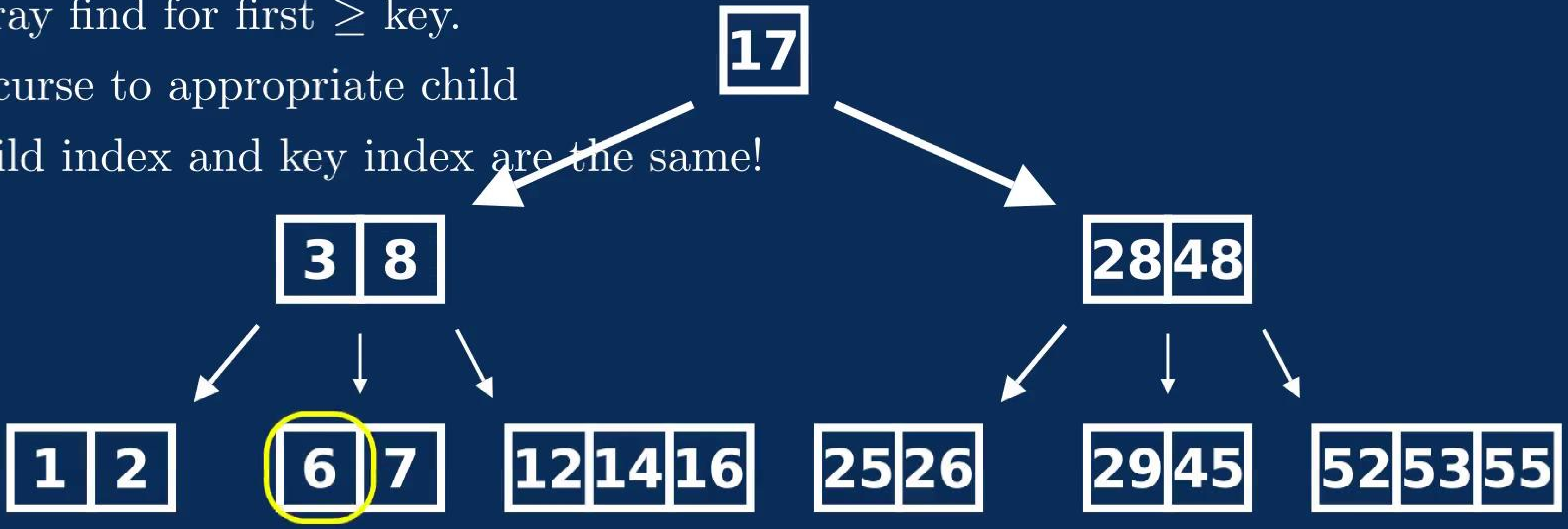
Base Case

- root empty ⇒ return
- leaf ⇒ do array find
-

Recursive Case

- Array find for first $\geq$ key.
- Recurse to appropriate child
- Child index and key index are the same!

find(6)



Find (Review)

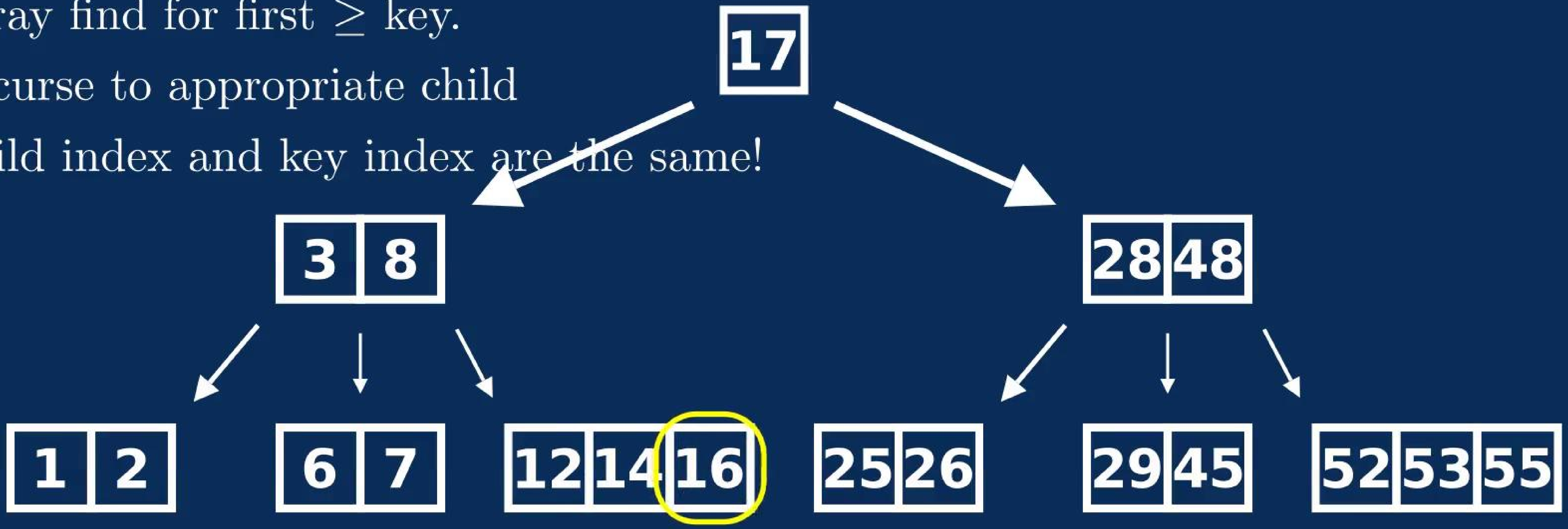
Base Case

- root empty $\Rightarrow$ return
- leaf $\Rightarrow$ do array find

Recursive Case

- Array find for first $\geq$ key.
- Recurse to appropriate child
- Child index and key index are the same!

find(16)



Find (Review)

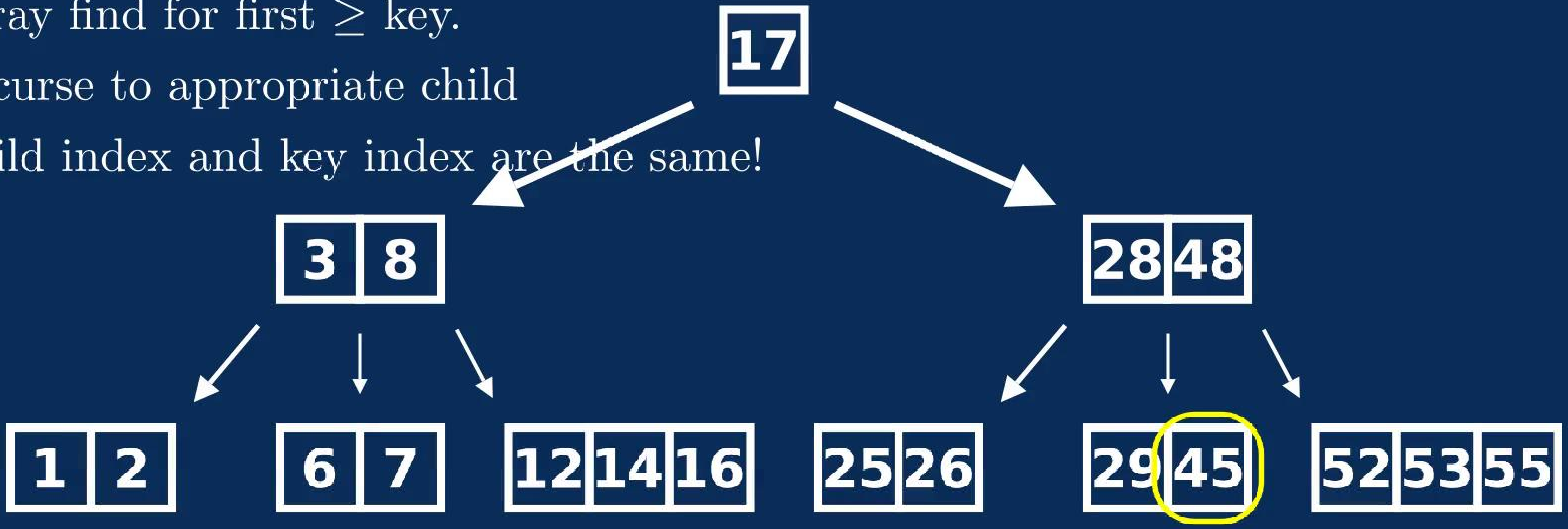
Base Case

- root empty ⇒ return
- leaf ⇒ do array find
-

Recursive Case

- Array find for first ≥ key.
- Recurse to appropriate child
- Child index and key index are the same!

find(45)



Array Insert

Node insert is simply ordered array insert.

Insert 5, 3, 8, 2, 4

Technically $\mathcal{O}(m)$ but **we don't care**.

- m is constant
- cost of seeking far more expensive

0	1	2	3	4
2	**3**	**4**	**5**	**8**

Splitting

Node full? Split the node and promote the median.

Let's suppose $m = 5$ and we just inserted 8.

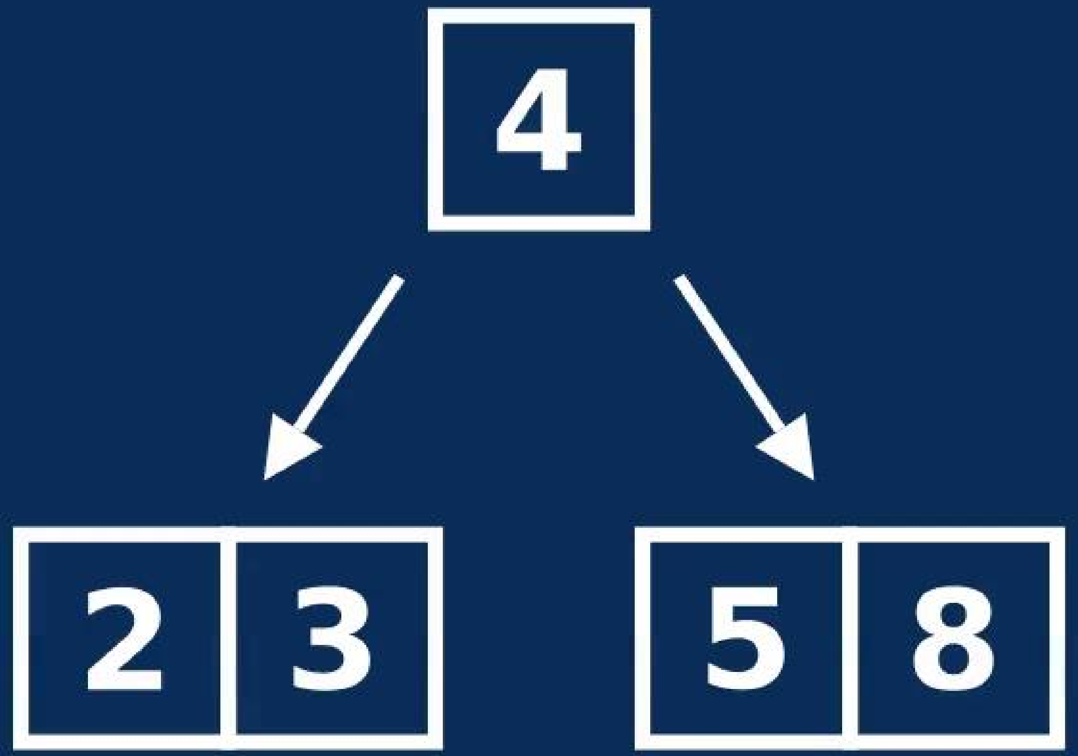
2	3	4	5	8

Splitting

Node full? Split the node and promote the median.

Let's suppose $m = 5$ and we just inserted 8.

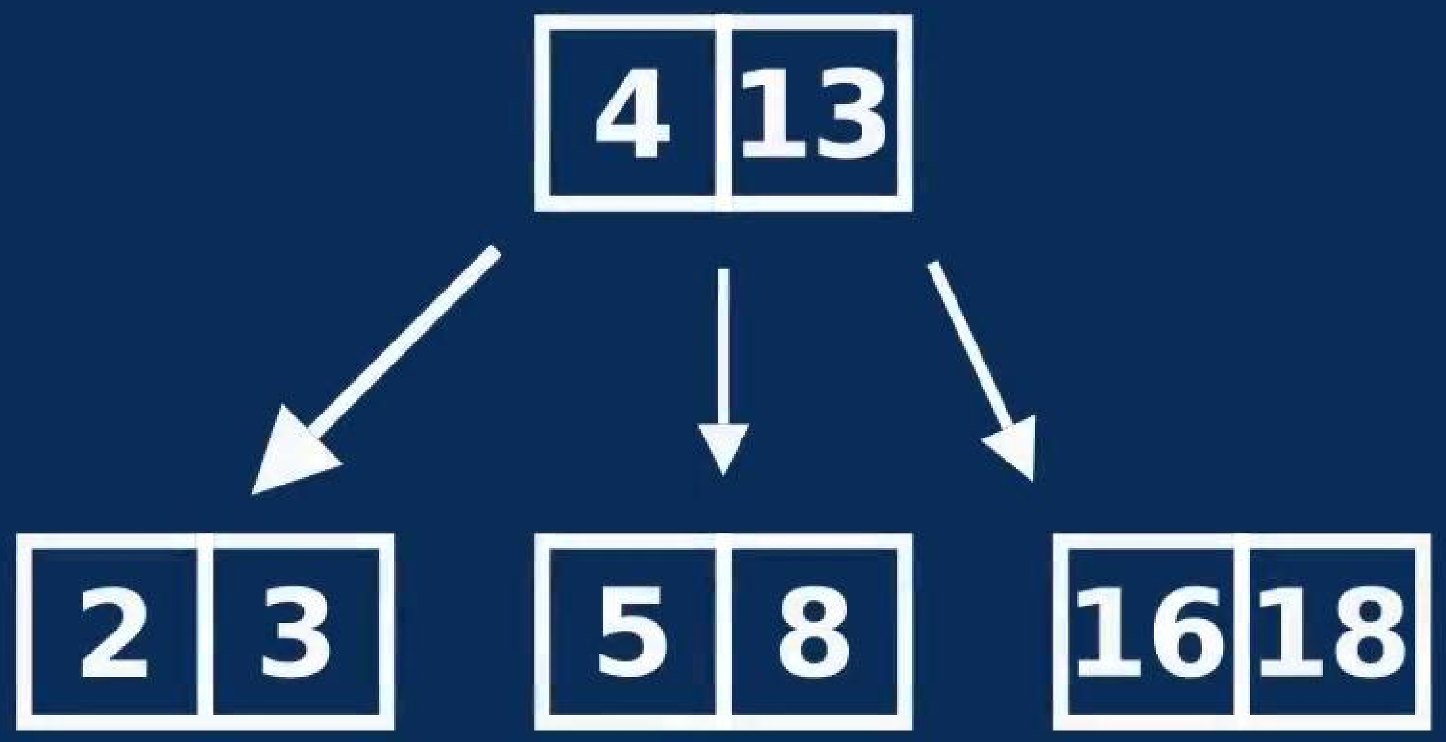
Split size (children) is $\lceil \frac{m}{2} \rceil$



Splitting

Splitting can happen recursively.

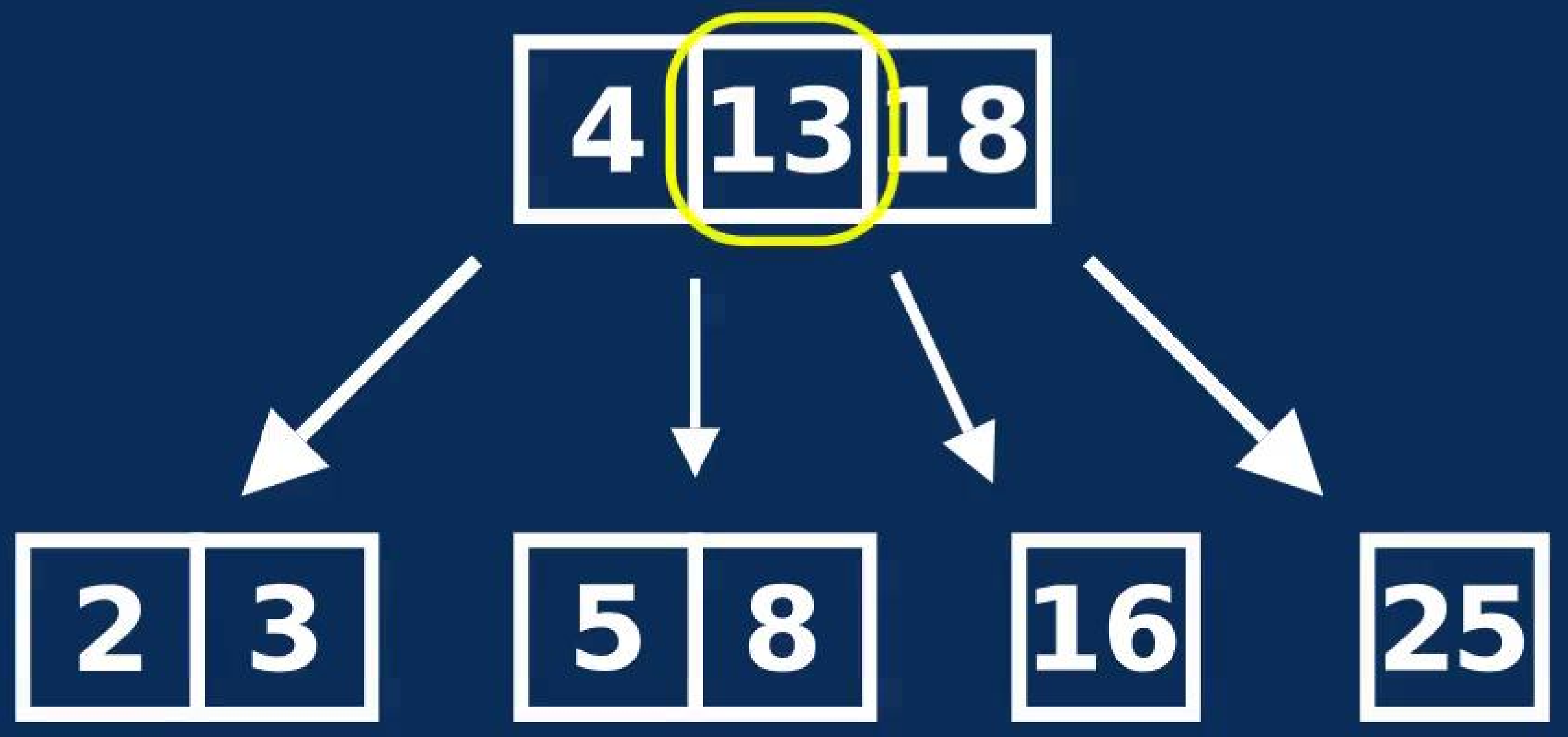
Let $m = 3$. Insert 25.



Splitting

Splitting can happen recursively.

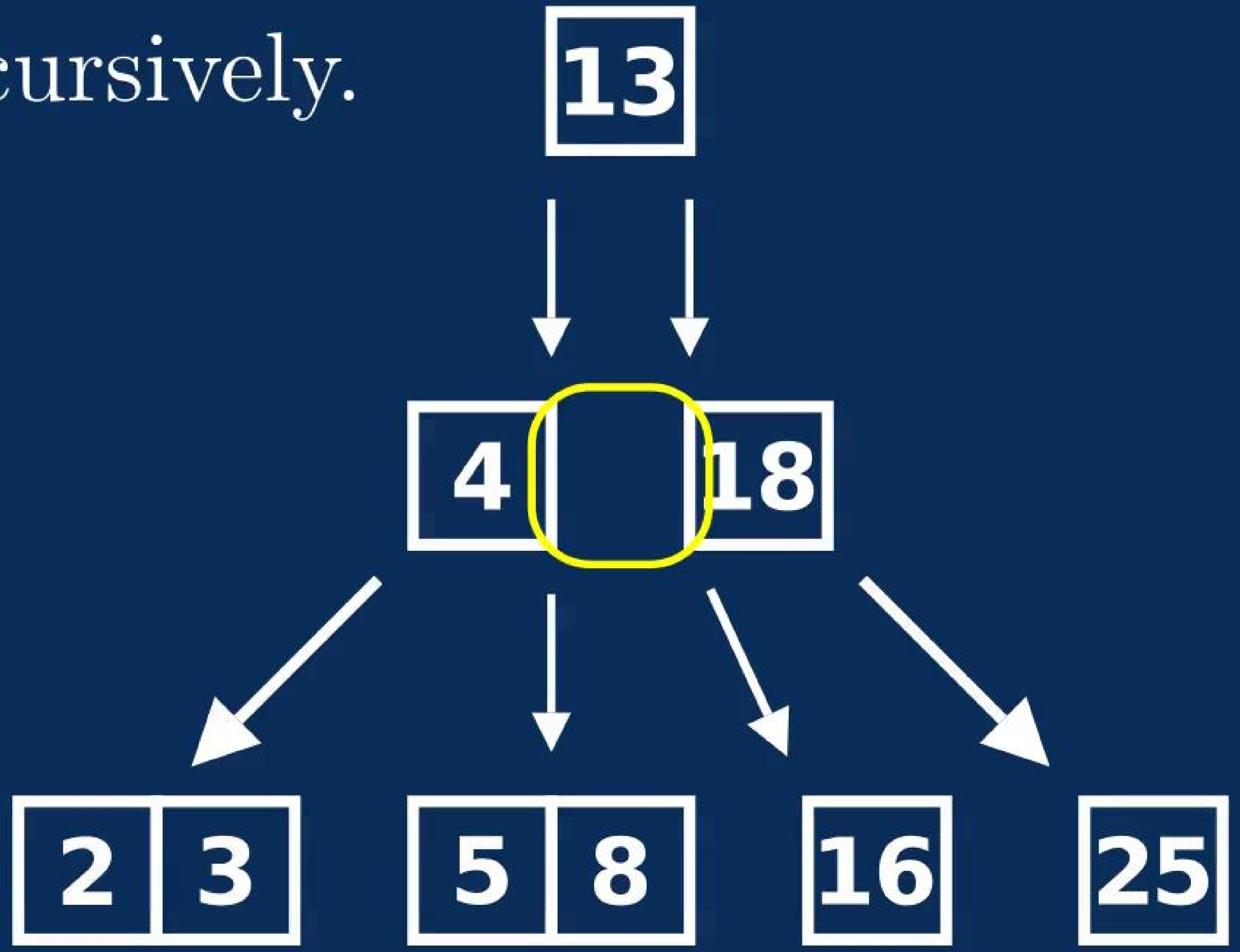
Let $m = 3$. Insert 25.



Splitting

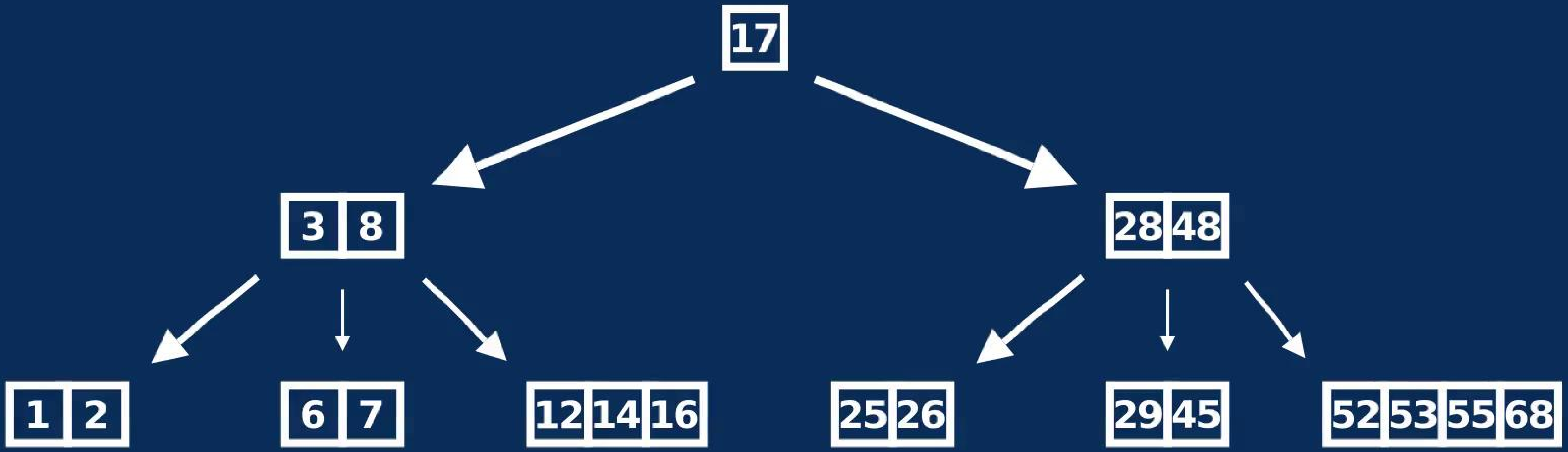
Splitting can happen recursively.

Let $m = 3$. Insert 25.



Lower Bound for m

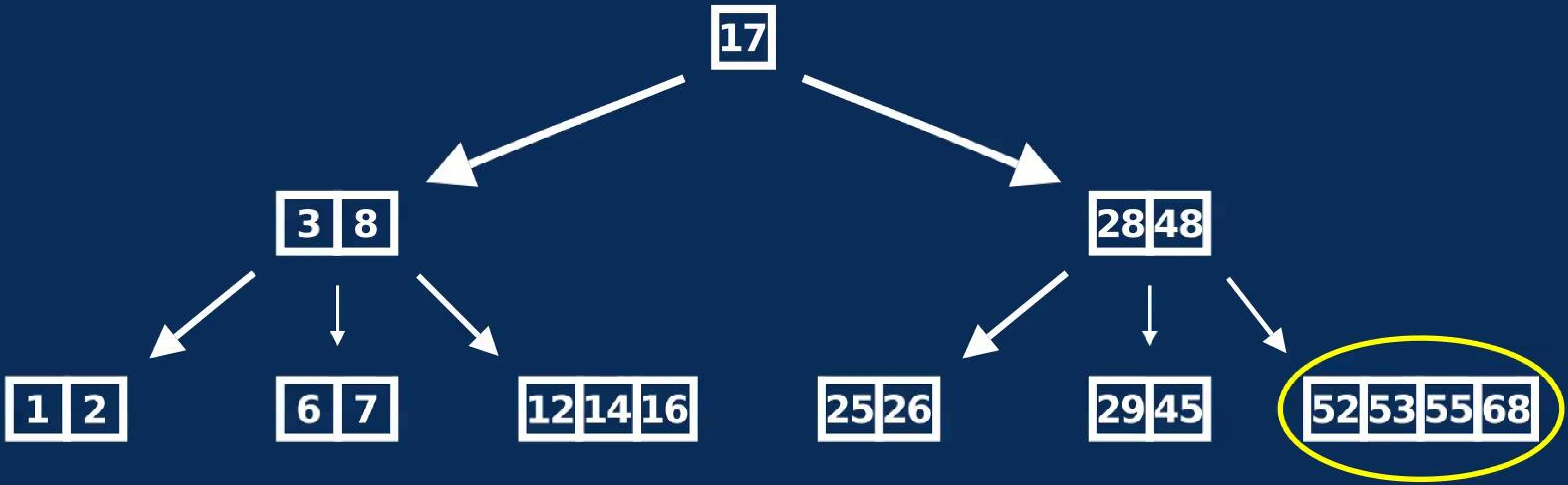
This is a valid B Tree. What is the lower bound for m?



Lower Bound for m

This is a valid B Tree. What is the lower bound for m?

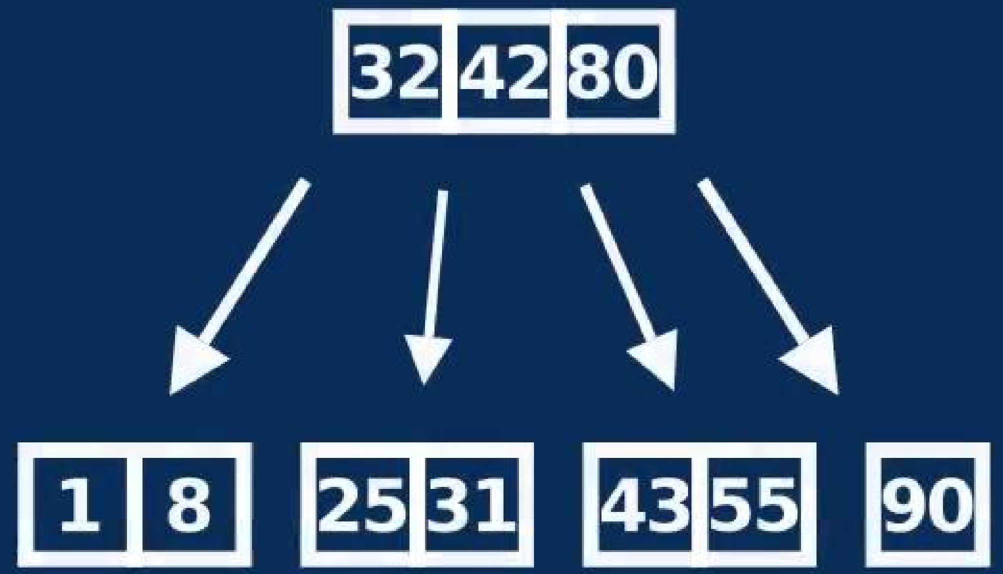
4 keys implies $m \leq 5$



Minimum node size

We have a max on keys. How about a min?

Is this tree valid? Let $m = 5$. Hint: $\lceil \frac{m}{2} \rceil = 3$

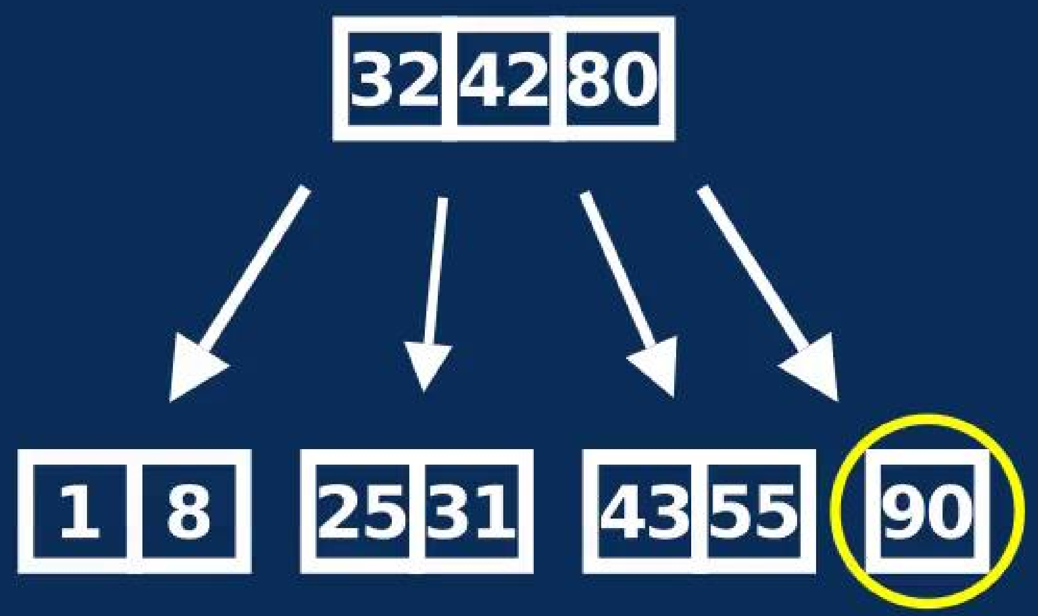


Minimum node size

We have a max on keys. How about a min?

Is this tree valid? Let $m = 5$. Hint: $\lceil \frac{m}{2} \rceil = 3$

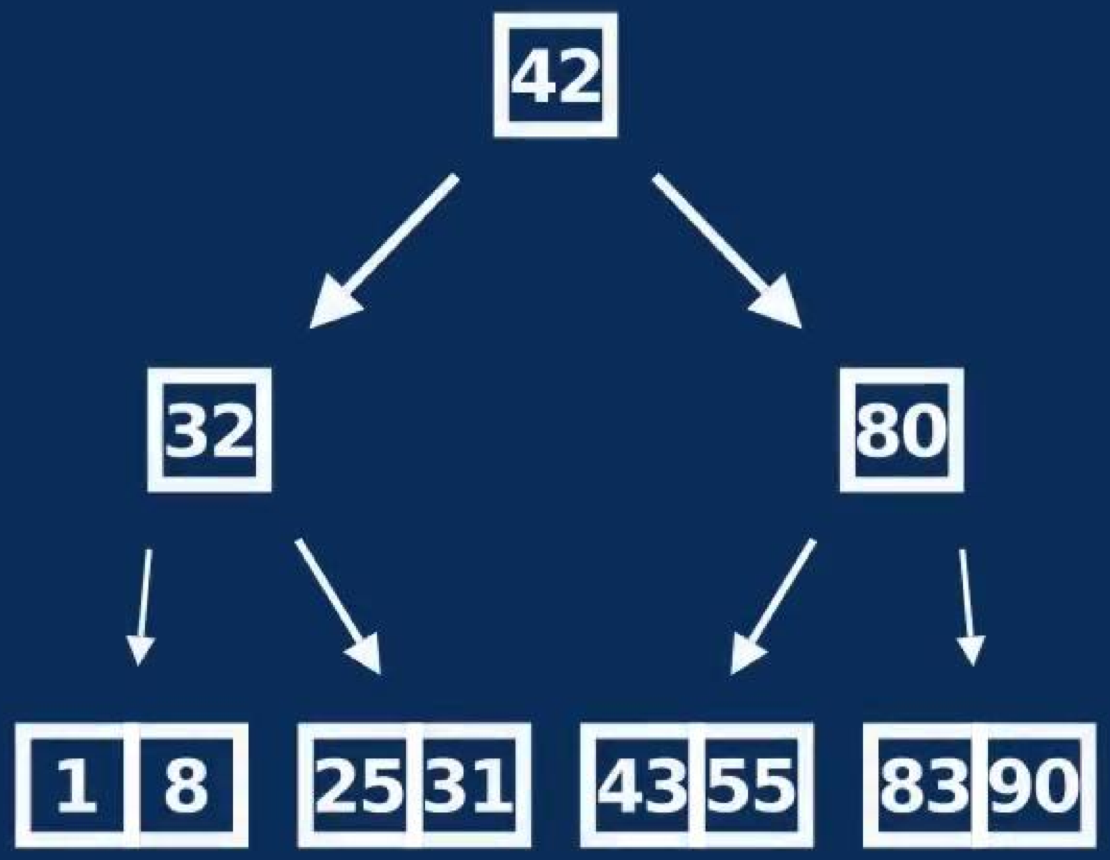
The bottom right leaf has $2 < \lceil \frac{m}{2} \rceil$ children



Minimum node size 2

We have a max on keys. How about a min?

Is this tree valid? Let $m = 5$. Hint: $\lceil \frac{m}{2} \rceil = 3$

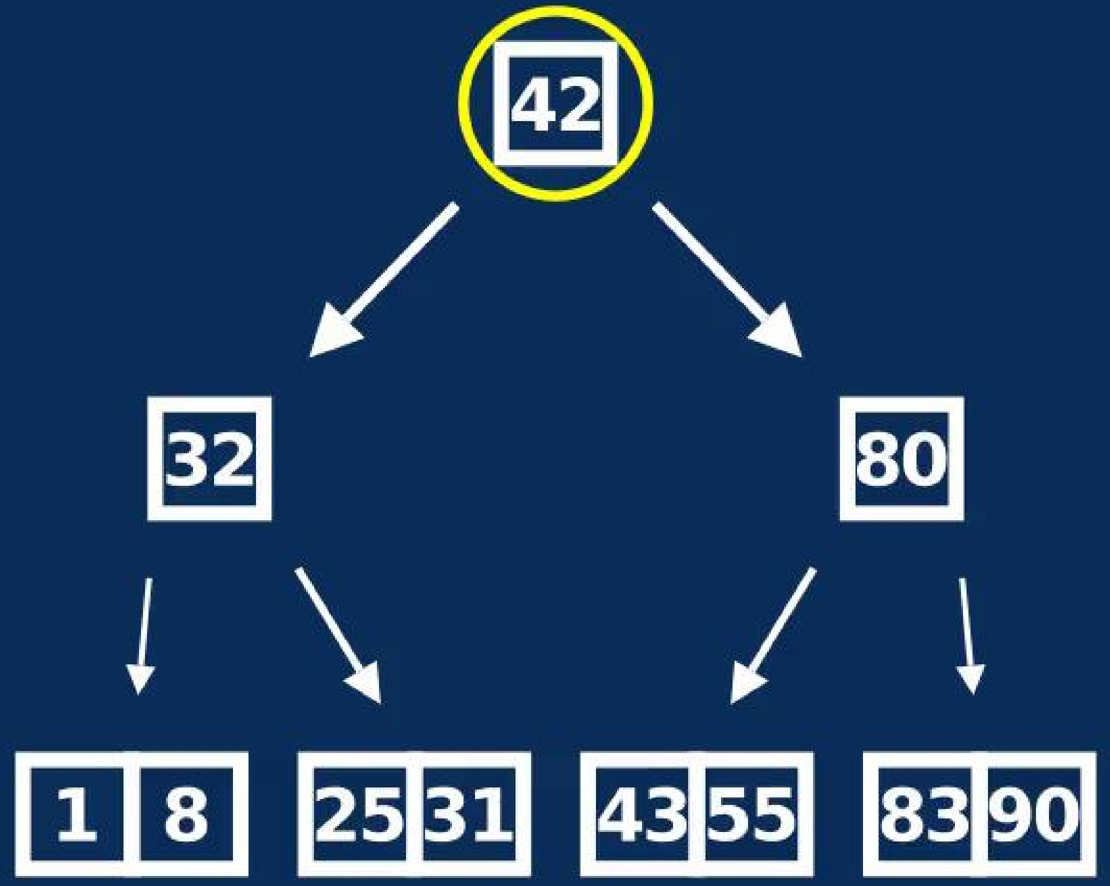


Minimum node size 2

We have a max on keys. How about a min?

Is this tree valid? Let $m = 5$. Hint: $\lceil \frac{m}{2} \rceil = 3$

The 32-42-80 node should not have been split.



Properties

A B Tree of order m is an m-ary tree

- All keys are ordered
- A node contains no more than $m-1$ keys
- Internal nodes have exactly **one more child than keys**
- All leaves are on the same level

Bounds on node sizes:

Properties

A B Tree of order m is an m-ary tree

- All keys are ordered
- A node contains no more than $m - 1$ keys
- Internal nodes have exactly **one more child than keys**
- All leaves are on the same level

Bounds on node sizes:

- Root nodes can be a leaf or have $[2, m]$ children.

Properties

A B Tree of order m is an m-ary tree

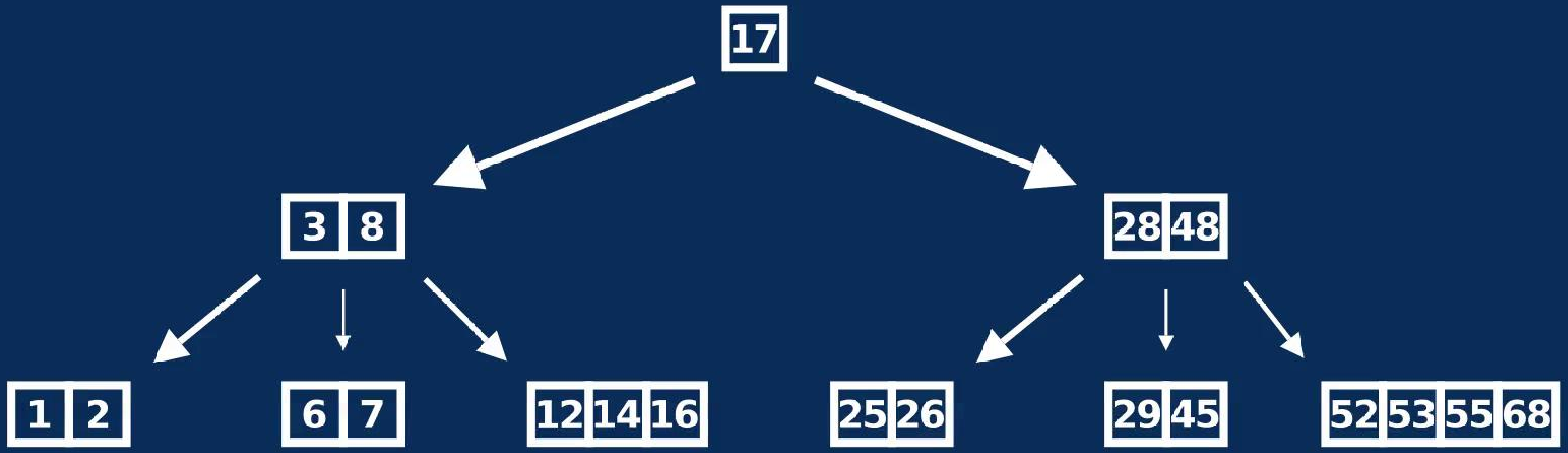
- All keys are ordered
- A node contains no more than $m-1$ keys
- Internal nodes have exactly **one more child than keys**
- All leaves are on the same level

Bounds on node sizes:

- Root nodes can be a leaf or have $[2, m]$ children.
- Non-root internal nodes have $[\lceil \frac{m}{2} \rceil, m]$ children.

Lower Bound for m

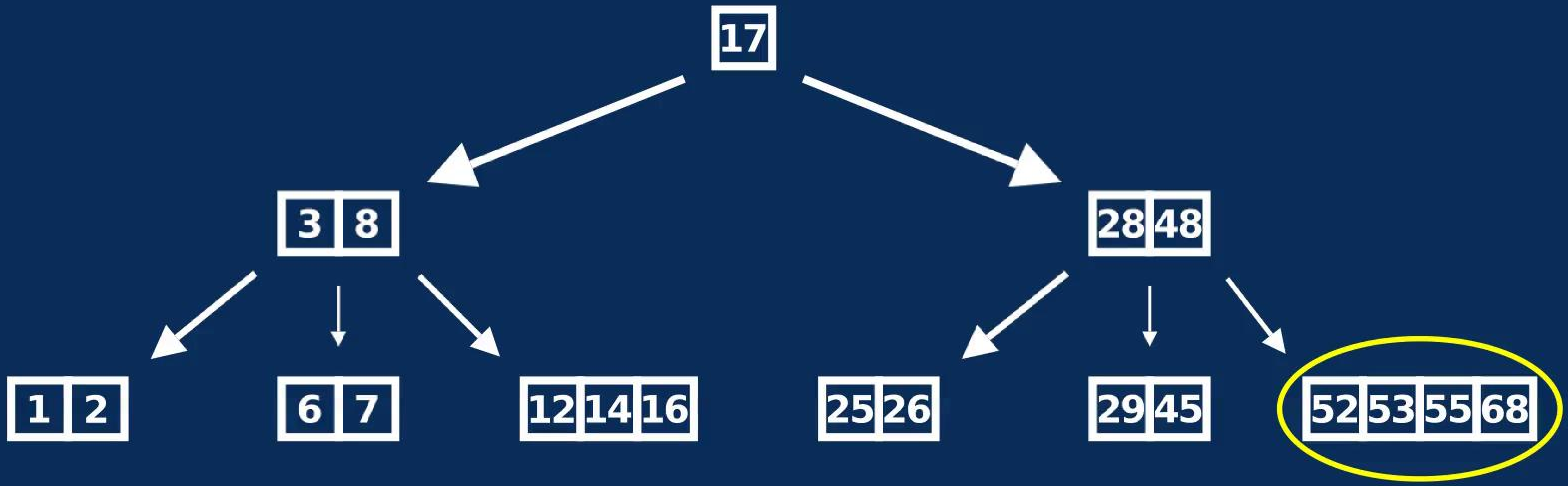
This is a valid B Tree. What is the precise value for m?



Lower Bound for m

This is a valid B Tree. What is the precise value for m?

4 keys implies $m \leq 5$

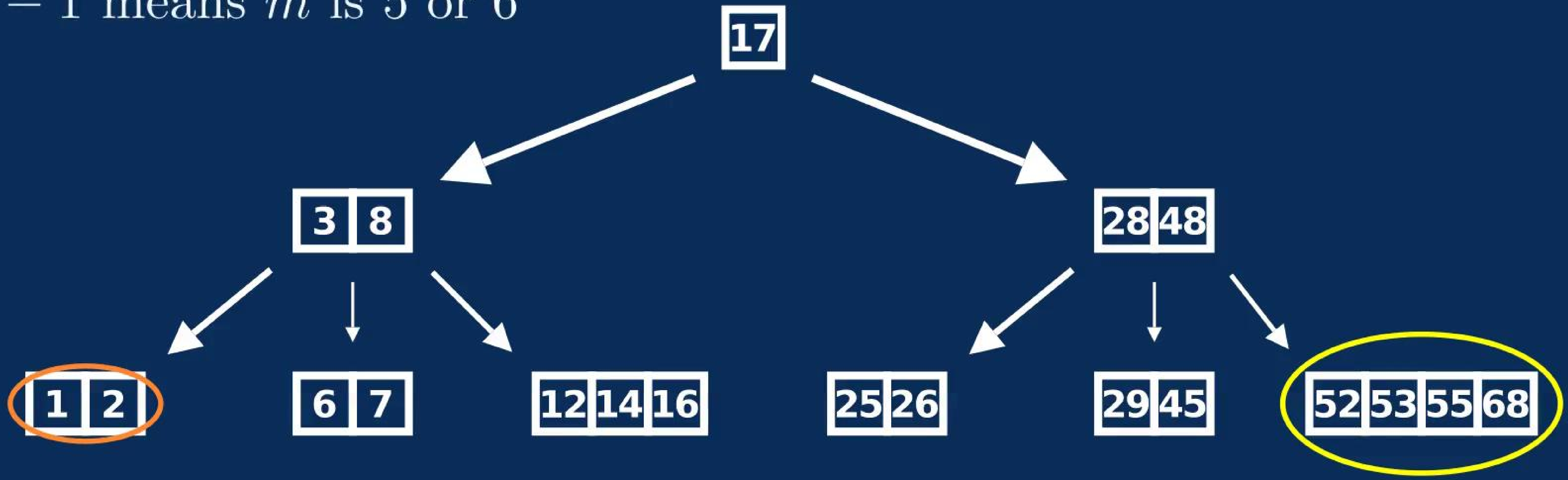


Lower Bound for m

This is a valid B Tree. What is the precise value for m?

4 keys implies $m \leq 5$

$\lceil \frac{m}{2} \rceil - 1$ means m is 5 or 6



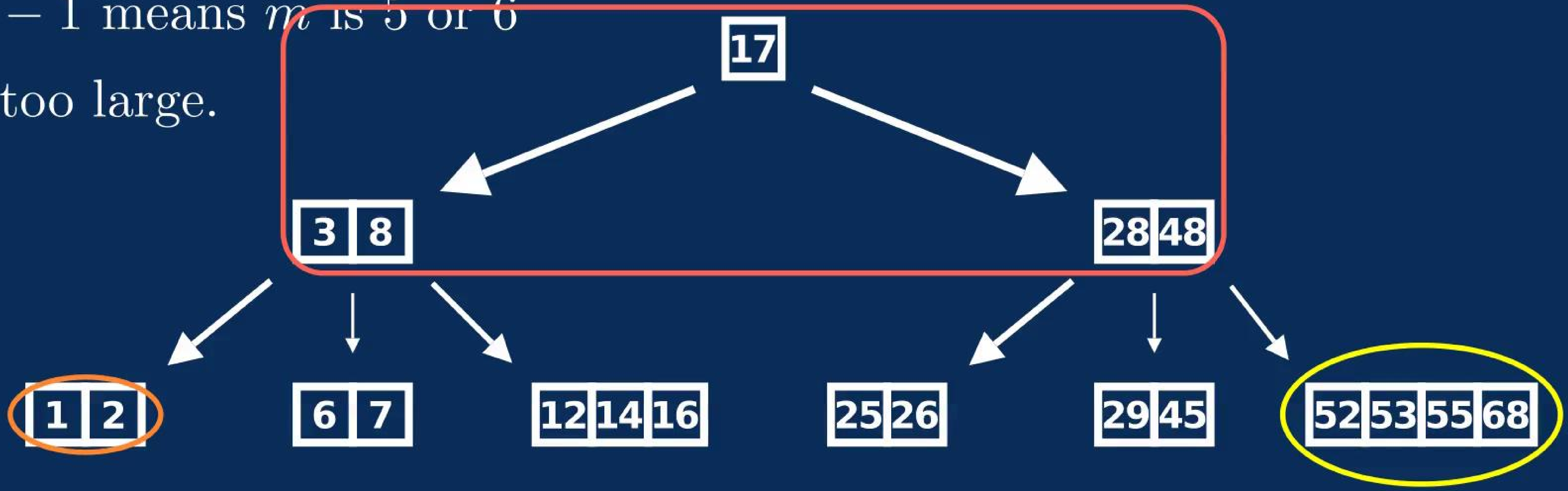
Lower Bound for m

This is a valid B Tree. What is the precise value for m?

4 keys implies $m \leq 5$

$\lceil \frac{m}{2} \rceil - 1$ means m is 5 or 6

6 is too large.



To Prove

B Tree height will determine runtime

Claim: B Tree limits height to $\mathcal{O}(log_m(n))$

Proof: find relationship between keys (n) and height (h)

Strategy

Count how many nodes are on each level

Add minimum number of keys per node to get n

This will tell us the largest possible height h

Minimum Nodes per Level

Let $t = \lceil \frac{m}{2} \rceil$

Think about number of nodes for root, levels 1,2,3,h

Minimum Nodes per Level

Let $t = \lceil \frac{m}{2} \rceil$

Think about number of nodes for root, levels 1,2,3,h

- Root: 1

Minimum Nodes per Level

Let $t = \lceil \frac{m}{2} \rceil$

Think about number of nodes for root, levels 1,2,3,h

- Root: 1
- Level 1: 2

Minimum Nodes per Level

Let $t = \lceil \frac{m}{2} \rceil$

Think about number of nodes for root, levels 1,2,3,h

- Root: 1
- Level 1: 2
- Level 2: $2t$

Minimum Nodes per Level

Let $t = \lceil \frac{m}{2} \rceil$

Think about number of nodes for root, levels 1,2,3,h

- Root: 1
- Level 1: 2
- Level 2: $2t$
- Level 3: $2t^2$

Minimum Nodes per Level

Let $t = \lceil \frac{m}{2} \rceil$

Think about number of nodes for root, levels 1,2,3,h

- Root: 1
- Level 1: 2
- Level 2: $2t$
- Level 3: $2t^2$
- Level h: $2t^{h-1}$

Min total Nodes

Minimum total nodes is $1 + 2\sum_{k=0}^{h-1} t^k$

Useful identity: $\sum_{i=0}^{n-1} x^i = \frac{x^n - 1}{x - 1}$

So $1 + 2\sum_{k=0}^{h-1} t^k = 1 + 2\frac{t^h - 1}{t - 1}$

Min total keys

Min total nodes is $1 + 2\sum_{k=0}^{h-1} t^k = 1 + 2\frac{t^h - 1}{t - 1}$

Minimum number of keys:

Root: 1 key / 1 node

Internal: $\lceil \frac{m}{2} \rceil - 1 = t - 1$

Leaf: $\lceil \frac{m}{2} \rceil - 1 = t - 1$

So we can multiply the fraction by $t - 1$!

Min total keys

Smallest number of keys is $2t^h - 1$, where $t = \lceil \frac{m}{2} \rceil$

Important inequality about n:

- $n \geq 2t^h - 1$

Min total keys

Smallest number of keys is $2t^h - 1$, where $t = \lceil \frac{m}{2} \rceil$

Important inequality about n:

- $n \geq 2t^h - 1$

Add one to both sides: $n + 1 \geq 2t^h$

Min total keys

Smallest number of keys is $2t^h - 1$, where $t = \lceil \frac{m}{2} \rceil$

Important inequality about n:

- $n \geq 2t^h - 1$

Add one to both sides: $n + 1 \geq 2t^h$

Take logs: $log_m(n+1) \geq log_m(2\lceil \frac{m}{2} \rceil^h)$

Min total keys

Smallest number of keys is $2t^h - 1$, where $t = \lceil \frac{m}{2} \rceil$

Important inequality about n:

- $n \geq 2t^h - 1$

Add one to both sides: $n + 1 \geq 2t^h$

Take logs: $log_m(n + 1) \geq log_m(2\lceil \frac{m}{2} \rceil^h)$

Ignore ceiling and constants: $log_m(m^h) = h$

Min total keys

Smallest number of keys is $2t^h - 1$, where $t = \lceil \frac{m}{2} \rceil$

Important inequality about n:

- $n \geq 2t^h - 1$

Add one to both sides: $n + 1 \geq 2t^h$

Take logs: $log_m(n+1) \geq log_m(2\lceil \frac{m}{2} \rceil^h)$

Ignore ceiling and constants: $log_m(m^h) = h$

Finally: we have $h = \mathcal{O}(log_m n)$

Conclusion

Smallest number of keys is $2t^h - 1$, where $t = \lceil \frac{m}{2} \rceil$

$h = \mathcal{O}(log_m n)$

What are min and max for $m = 101$ and $h = 4$?

Conclusion

Smallest number of keys is $2t^h - 1$, where $t = \lceil \frac{m}{2} \rceil$

$h = \mathcal{O}(log_m n)$

What are min and max for $m = 101$ and $h = 4$?

- Min is $2 \times 51^4 - 1 = 13,530,401$

Conclusion

Smallest number of keys is $2t^h - 1$, where $t = \lceil \frac{m}{2} \rceil$

$h = \mathcal{O}(log_m n)$

What are min and max for $m = 101$ and $h = 4$?

- Min is $2 \times 51^4 - 1 = 13,530,401$
- Max is $100 + 101^1 \times 100 + 101^2 \times 100 + \cdots$

Conclusion

Smallest number of keys is $2t^h - 1$, where $t = \lceil \frac{m}{2} \rceil$

$h = \mathcal{O}(log_m n)$

What are min and max for $m = 101$ and $h = 4$?

- Min is $2 \times 51^4 - 1 = 13,530,401$
- Max is $100 + 101^1 \times 100 + 101^2 \times 100 + \cdots$
- $= 100 \sum_{i=0}^{4} 101^i$
- $= 100 \frac{101^5 - 1}{101 - 1} = 10,510,100,500$