Announcements

Join Code: **225**

Thanks for being a great class!

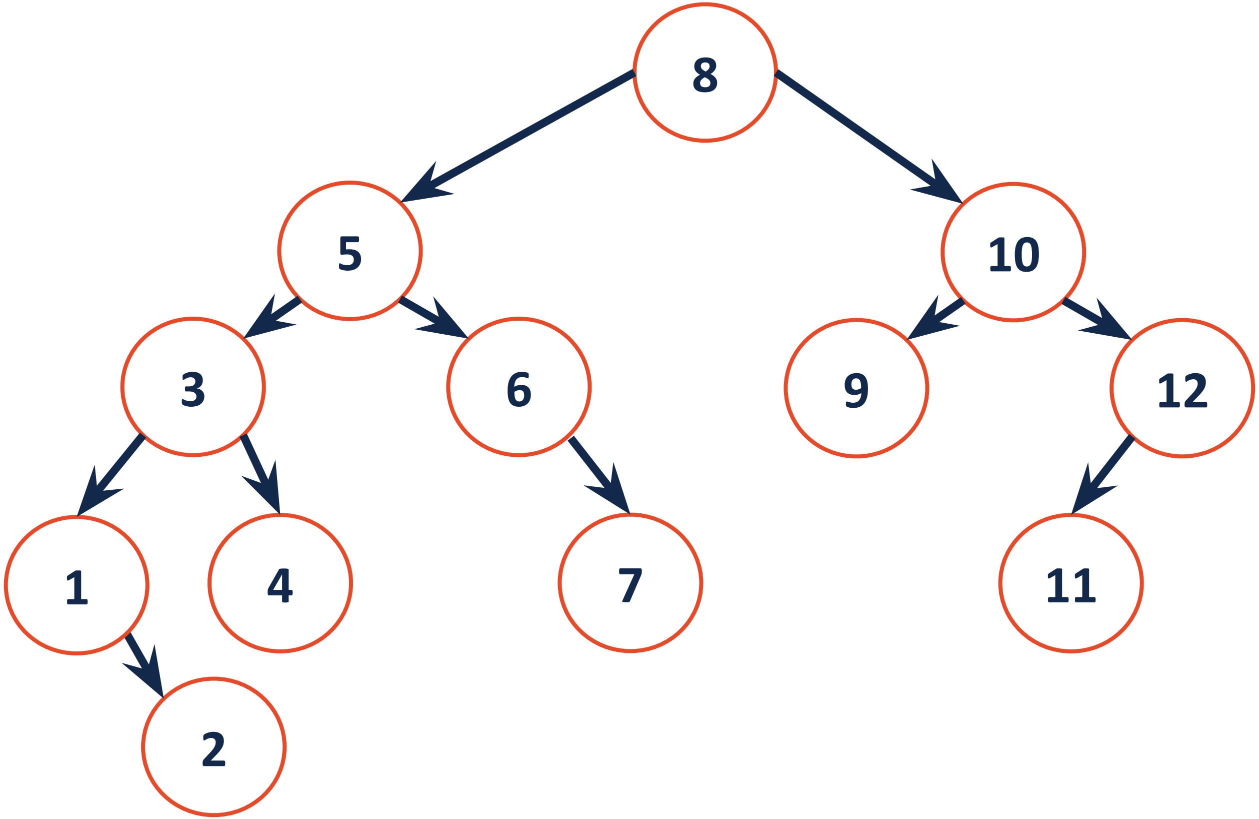
ILLINOIS

BTree Introduction

Learning Objectives

1. Understand the motivation behind BTrees
2. Know all the BTree Properties
3. Understand how to insert elements into a BTree
4. Implement Search in a BTree

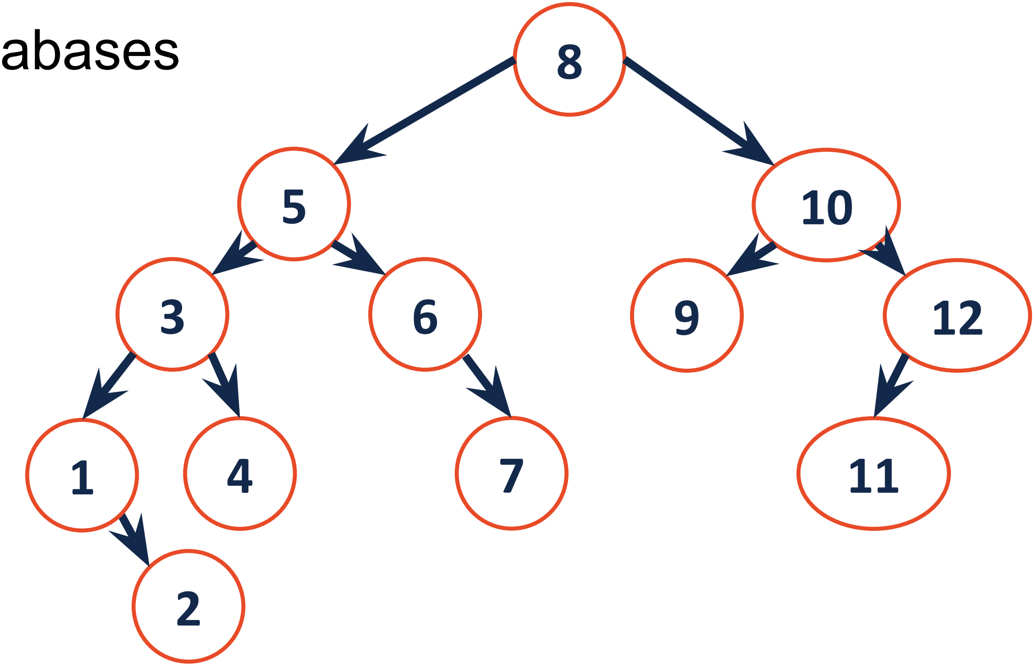
AVL Trees



Rate Limiting Factor - Seeks

Distributed Databases

Hard Disks



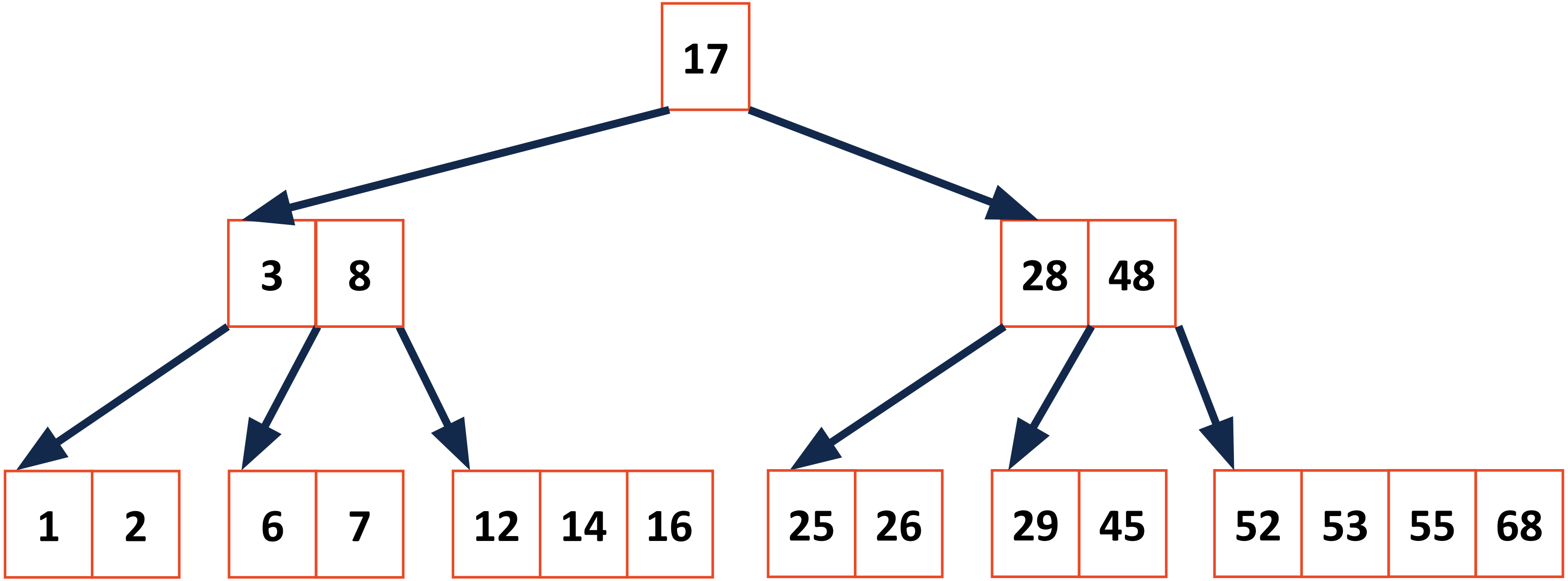
System Design

Goal: Minimize the number of reads!

Build a tree that uses ________________________ / node

1. Distributed Database - [1 network packet]
   a. 1500 Bytes (1 MTU) (Maximum Transmissible Unit)
2. Hard Disk - [1 disk block]
   a. Old Systems 512 Bytes
   b. New Systems 4096 Bytes

What are some observations about B-Trees?



BTree Properties

A **BTrees** of order **m** is an m-way tree:

Definition

- All keys within a node are ordered
- All nodes contain no more than **m-1** keys
- All internal nodes have **one more child than keys**
- All leaves are on the same level

B-Tree Operations

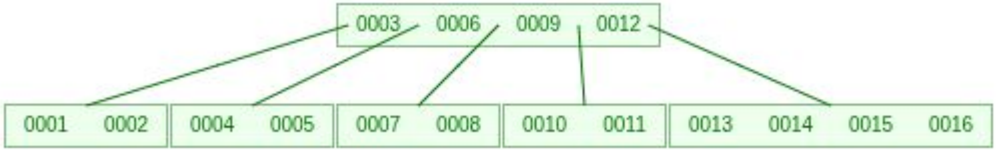
Insert

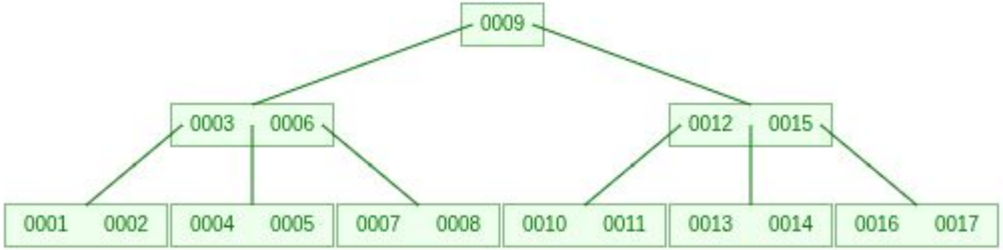
Find

Remove - More complicated, there is a POTD about this, not discussed in this class

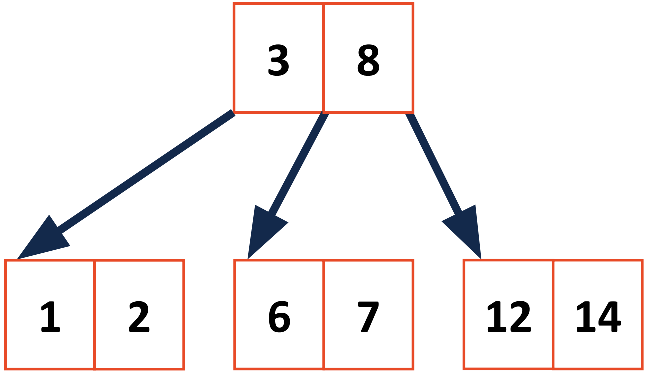
Insert - Recursive Split

Insert(17)



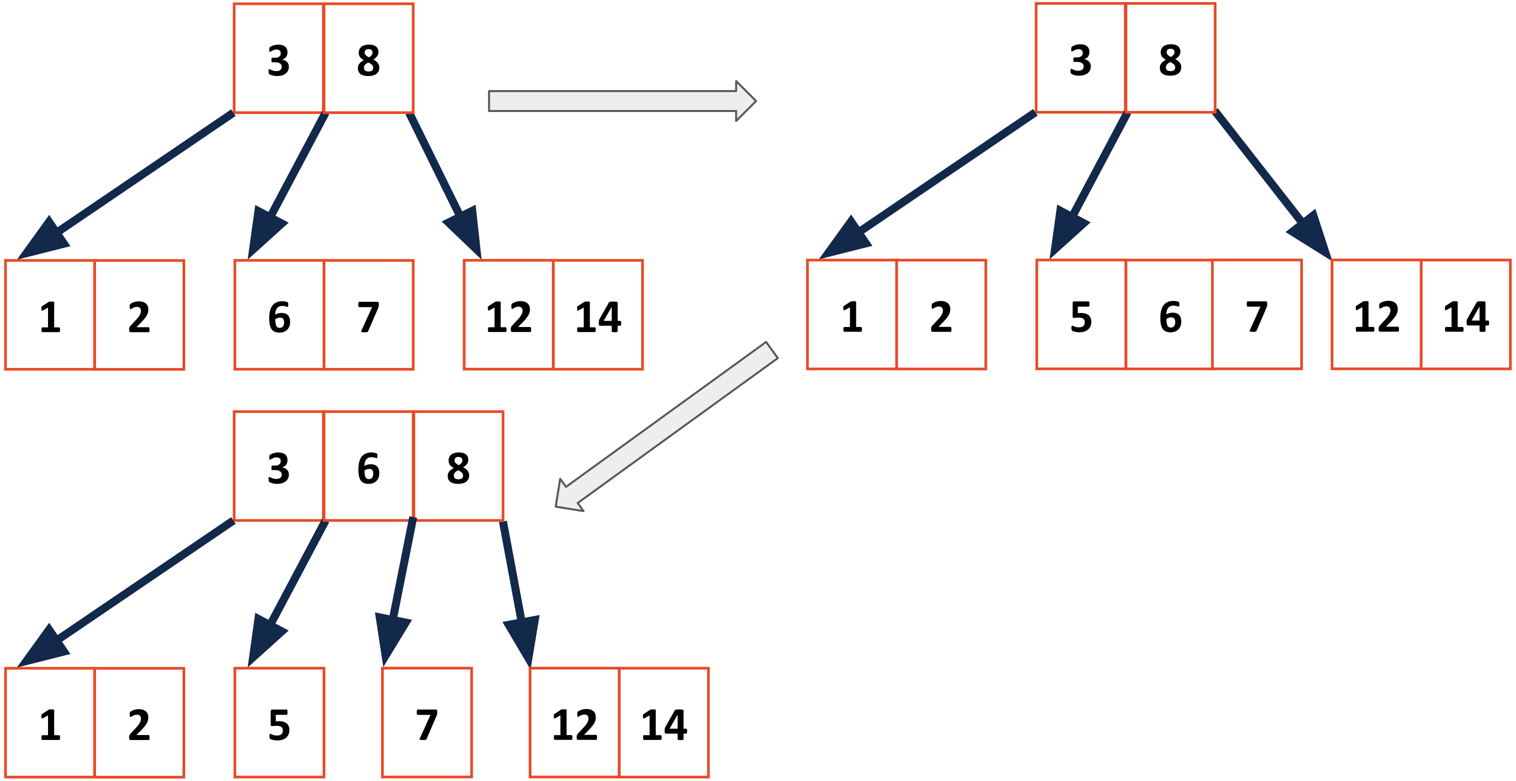


What will be the root after insert(5)? Assume m = 3

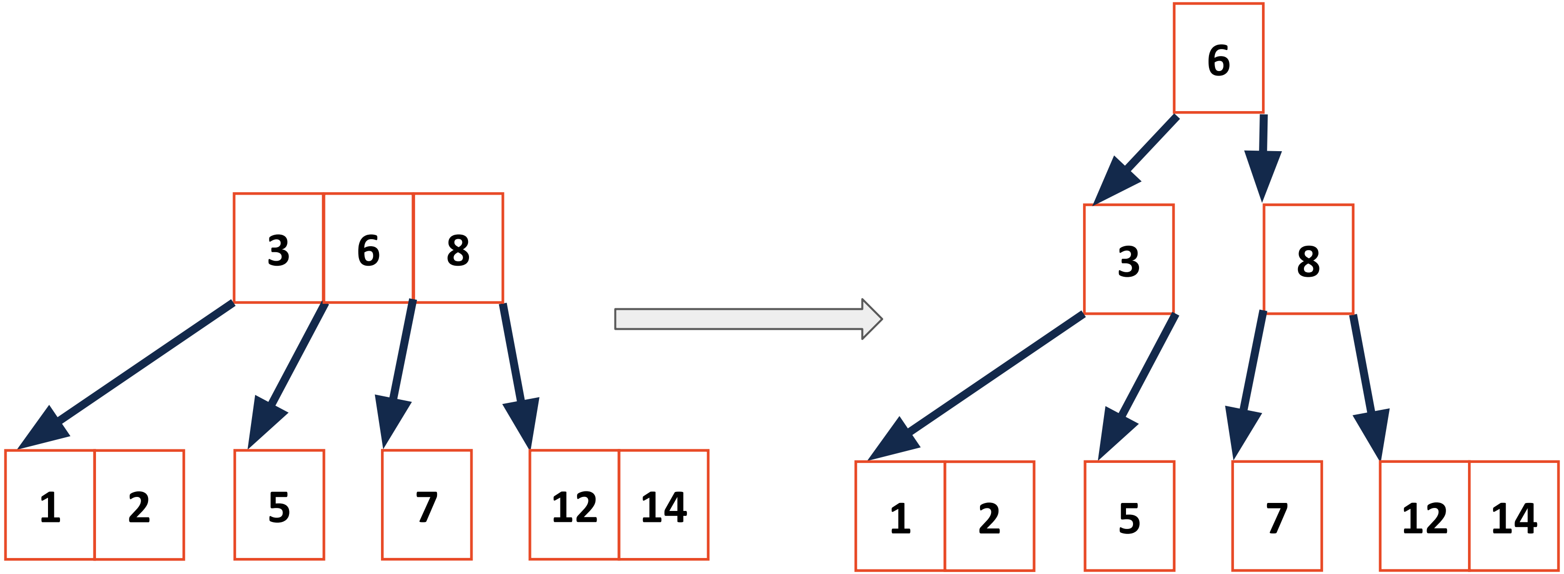


Join Code: **225**

What will be the root after insert(5)? Assume m = 3



What will be the root after insert(5)? Assume m = 3



B-Tree Insert

- Always inserts at leaf level
- Split if m nodes are reached
- Split could propagate up the tree recursively

BTree Properties

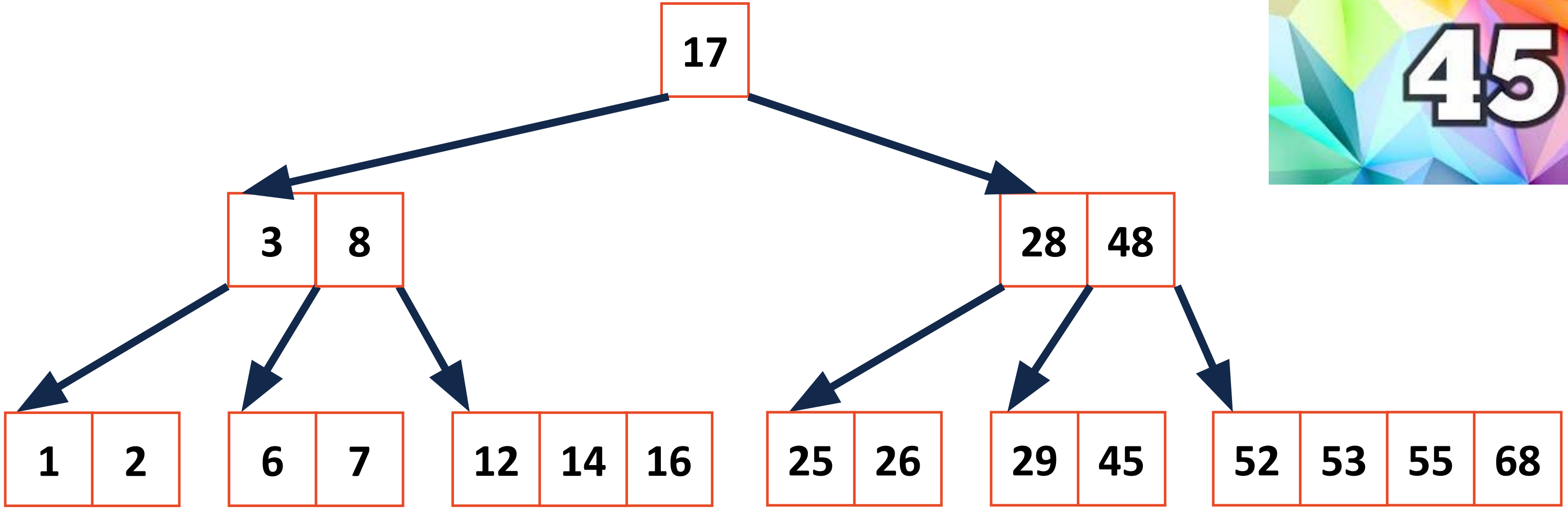
A **BTrees** of order **m** is an m-way tree:

Definition

- All keys within a node are ordered
- All nodes contain no more than **m-1** keys
- All internal nodes have **one more child than keys**
- All leaves are on the same level
- Root node is a leaf or has **[2, m]** children.
- All non-root, internal nodes have **[ceil(m/2), m]** children

If this is a valid BTree, what is the lower bound for m?

Join Code: **225**



17

3 | 8

28 | 48

1 | 2

6 | 7

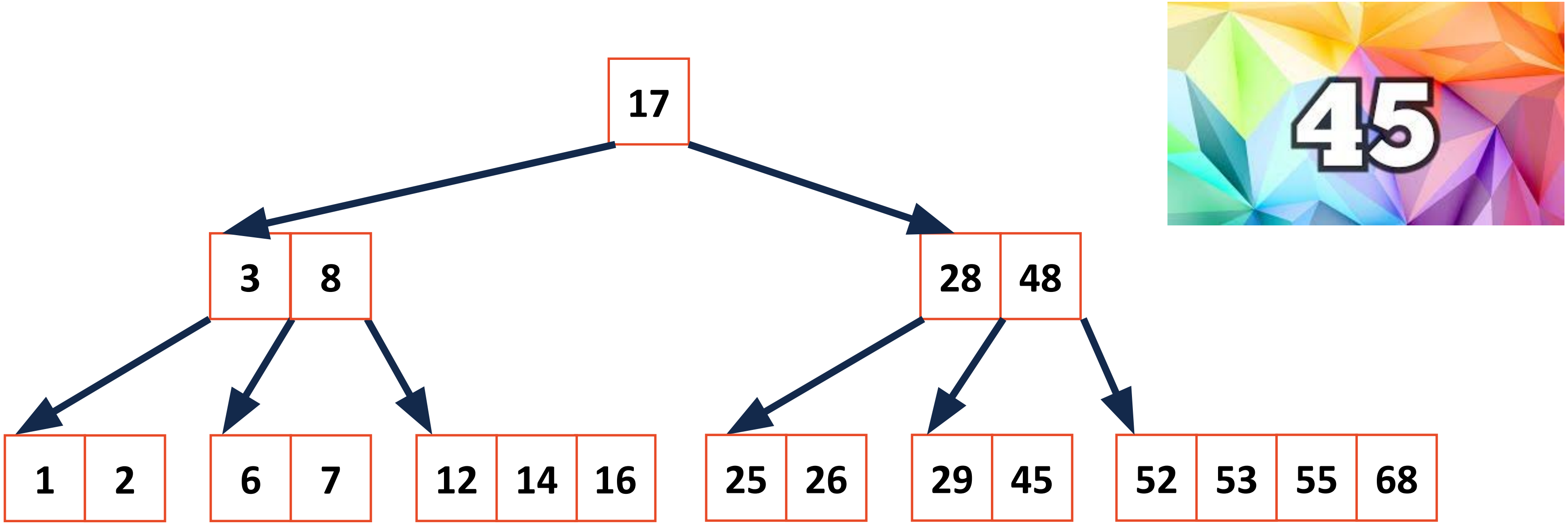
12 | 14 | 16

25 | 26

29 | 45

52 | 53 | 55 | 68

If this is a valid BTree, what is the upper bound for m?



17

3 8

28 48

1 2

6 7

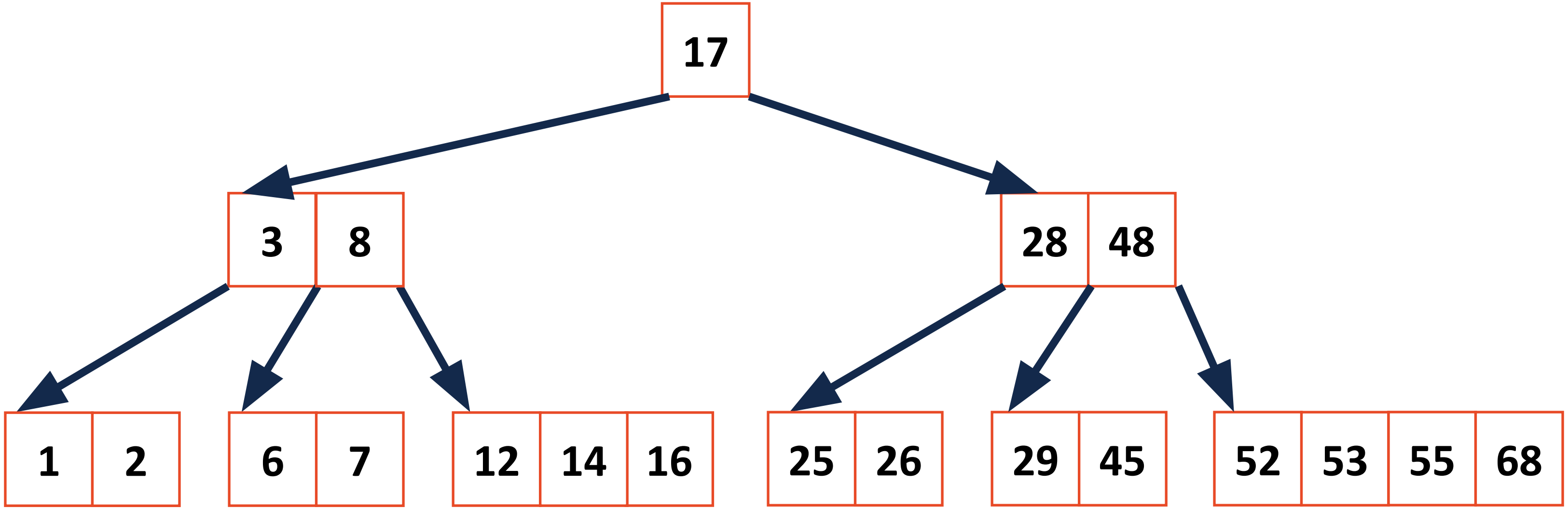
12 14 16

25 26

29 45

52 53 55 68

B-Tree Find: How would find(12) work? find(54)?



BTree Search

```
template <typename K, typename V>
V BTree<K, V>::_find(BTreeNode * node, const K & key) const {
  unsigned i;
  for (i = 0; i < node.keys_ct_ && key > node.keys_[i]; i++) { }

  if (i < node.keys_ct_ && key == node.keys_[i]) {
    return node.values_[i];
  }

  if (node.isLeaf()) {
    return V();
  }

  BTreeNode nextChild = node._fetchChild(i);
  return _find(nextChild, key);
}
```

BTree Properties

A **BTrees** of order **m** is an m-way tree:

Definition

- All keys within a node are ordered
- All nodes contain no more than **m-1** keys
- All internal nodes have **one more child than keys**
- All leaves are on the same level
- Root node is a leaf or has **[2, m]** children.
- All non-root, internal nodes have **[ceil(m/2), m]** children