

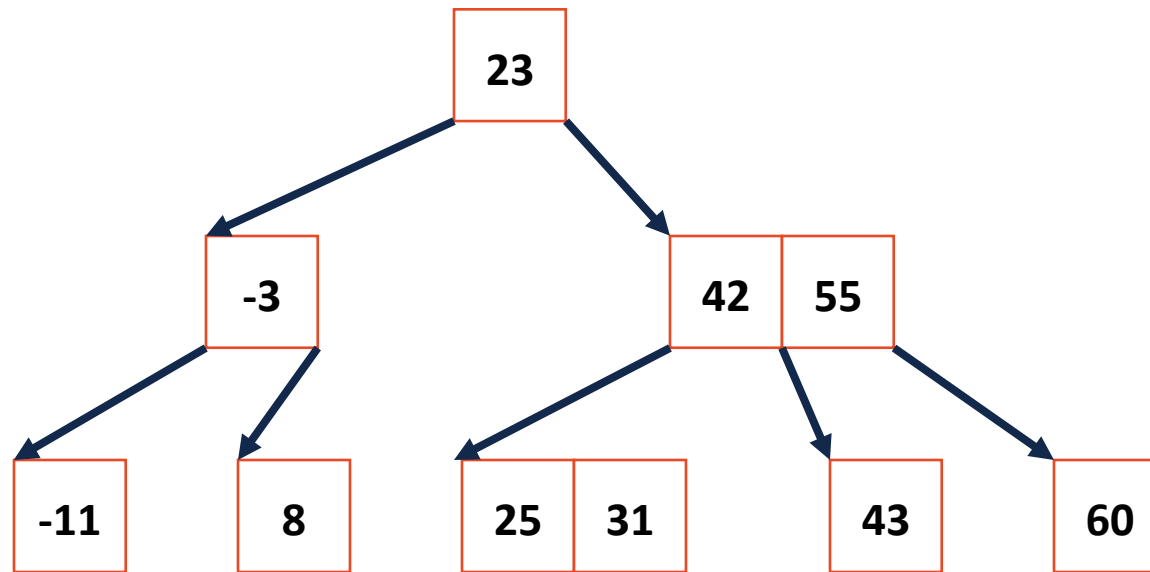


# CS 400

## **B-Tree Search**

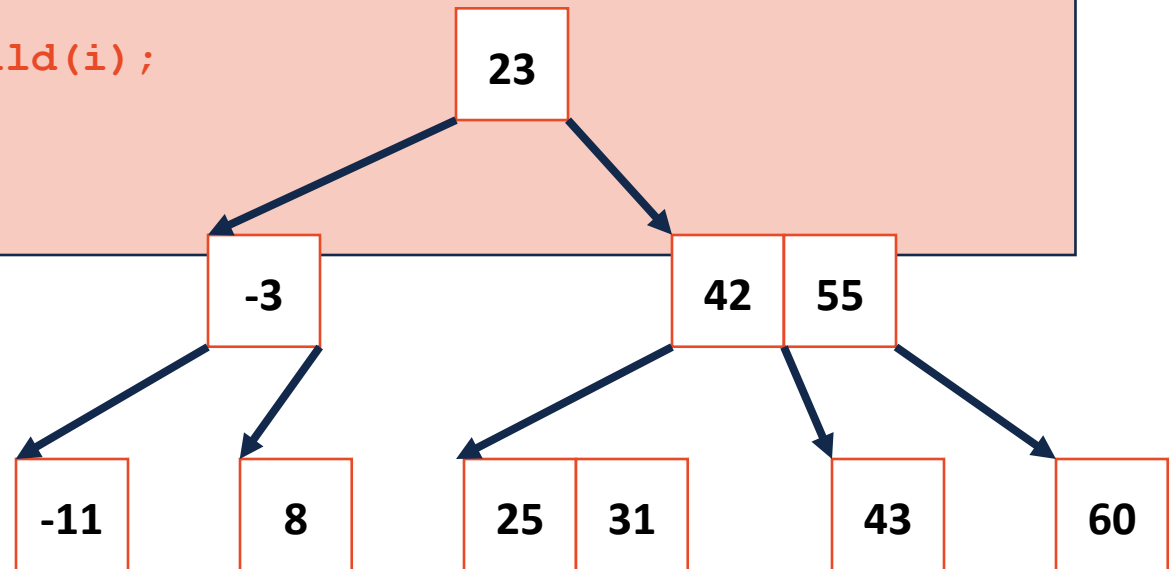
**ID: 08-03**

# BTree Search



# BTree Search

```
1 bool Btree::_exists(BTreeNode & node, const K & key) {  
2  
3     unsigned i;  
4     for ( i = 0; i < node.keys_ct_ && key < node.keys_[i]; i++) { }  
5  
6     if ( i < node.keys_ct_ && key == node.keys_[i] ) {  
7         return true;  
8     }  
9  
10    if ( node.isLeaf() ) {  
11        return false;  
12    } else {  
13        BTreeNode nextChild = node._fetchChild(i);  
14        return _exists(nextChild, key);  
15    }  
16 }
```



# BTree Analysis

The height of the BTree determines maximum number of  
      <sup>seeks</sup>       possible in search data.

...and the height of the structure is:       <sup>log<sub>m</sub>(n)</sup>      .

**Therefore:** The number of seeks is no more than       <sup>log<sub>m</sub>(n)</sup>      .