

附加内容：B 树

初航，我带你；远航，靠自己
胡船长

B 树 – 结构定义

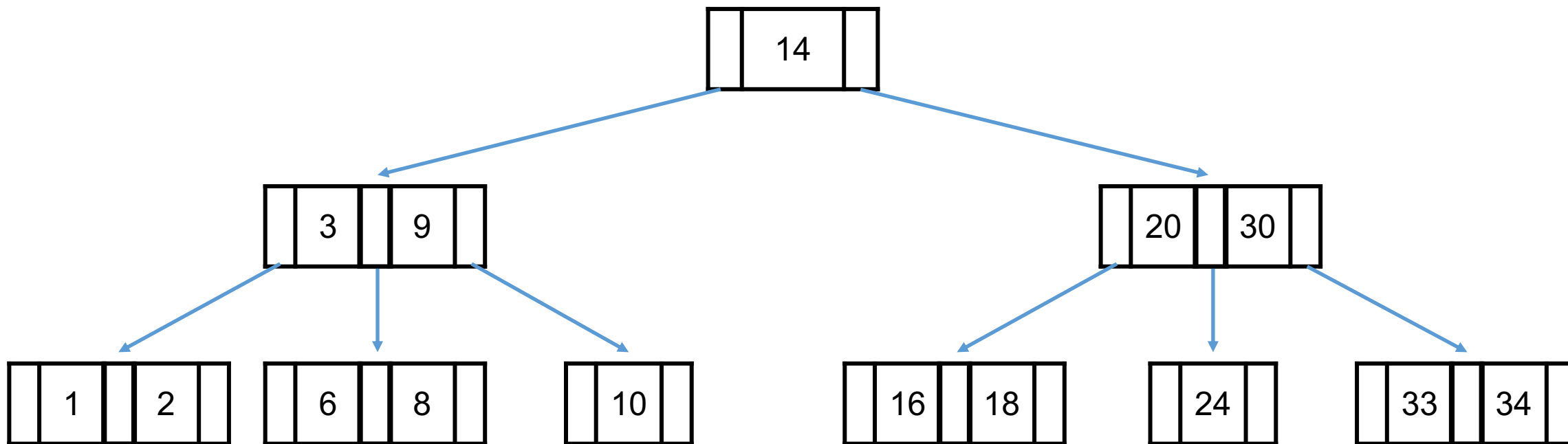
一棵 m 阶 B 树，需要满足下列特性：

1. 树中每个节点，最多含有 m 棵子树
2. 若根节点不是叶子节点，则至少有 2 棵子树
3. 除根结点之外的所有非终端结点至少有 $\lceil m/2 \rceil$ 棵子树
4. 如果一个结点有 $n-1$ 个关键字，则该结点有 n 个分支，且这 $n-1$ 个关键字按照递增顺序排列
5. 每个结点的结构为：($n, A_0, K_1, A_1, K_2, A_2, \dots, K_n, A_n$)
6. 非根结点中关键字的个数 n ，满足： $\lceil m/2 \rceil - 1 \leq n \leq m-1$
7. 所有叶子节点处在同一层

B 树 – 结构定义

m 阶 B 树的性质解读：

1. B 树中只有根节点没办法满足拥有至少 $\lceil m/2 \rceil$ 棵子树的条件，其他节点均能满足
2. B 树是一种高度平衡的树形结构，比 AVL 树结构上更优美





1. 元素插入

2. 插入调整

3. 元素删除

4. 删除调整





1. 元素插入

2. 插入调整



3. 元素删除

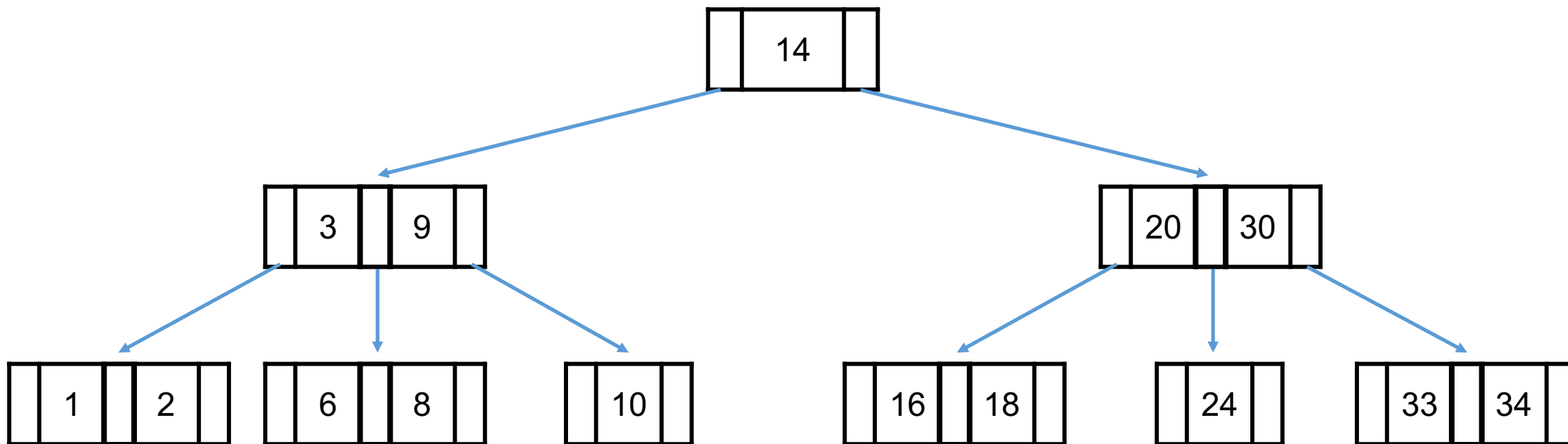
4. 删除调整



B 树 – 元素插入

插入:

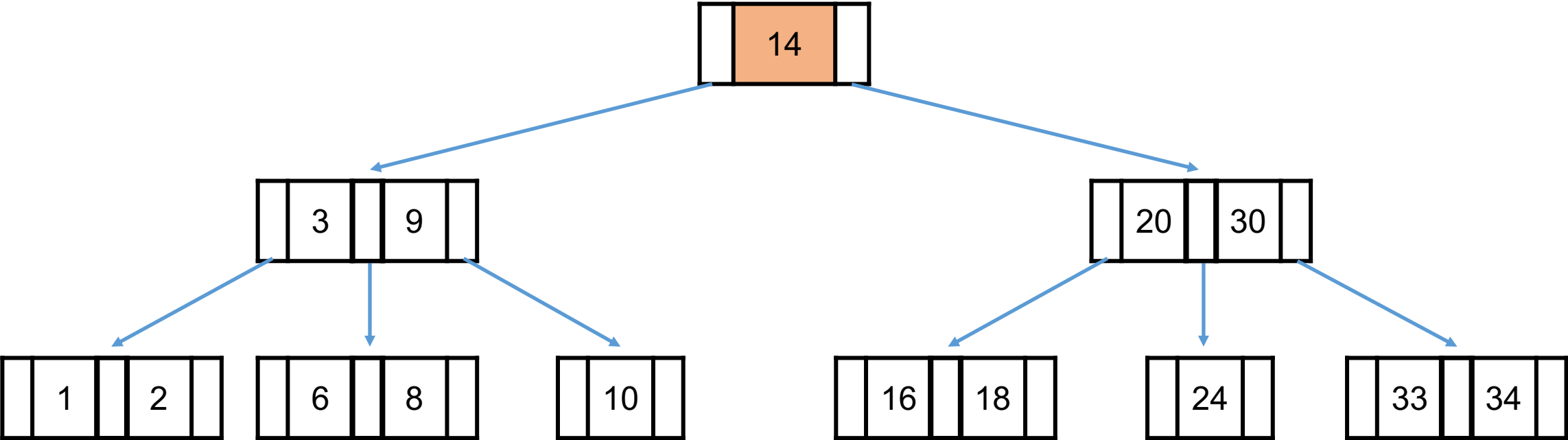
	26	
--	----	--



B 树 – 元素插入

插入:

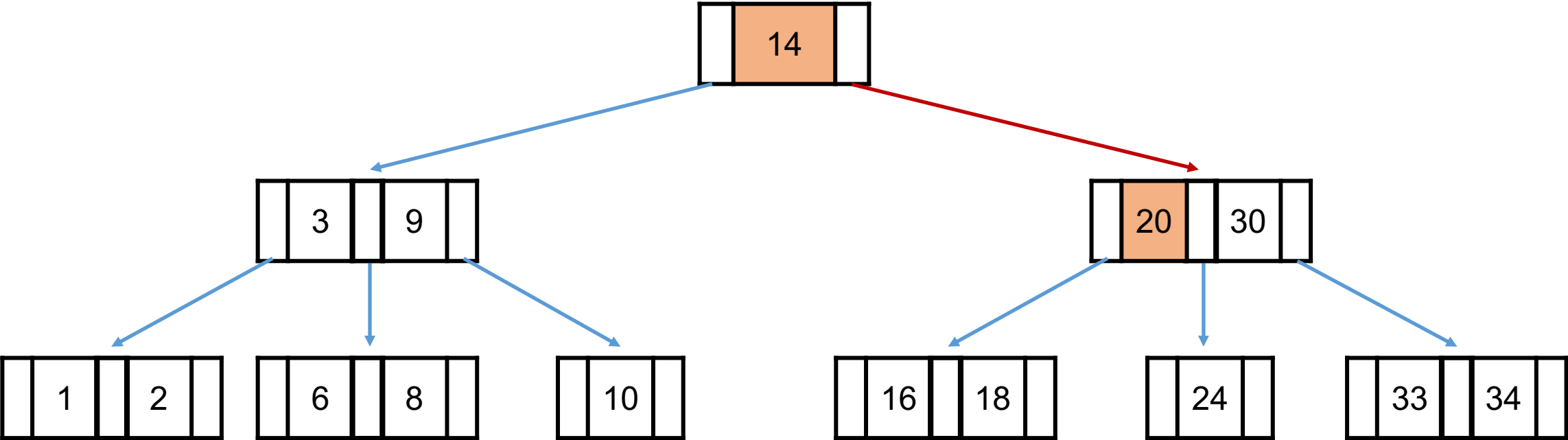
	26	
--	----	--



B 树 – 元素插入

插入:

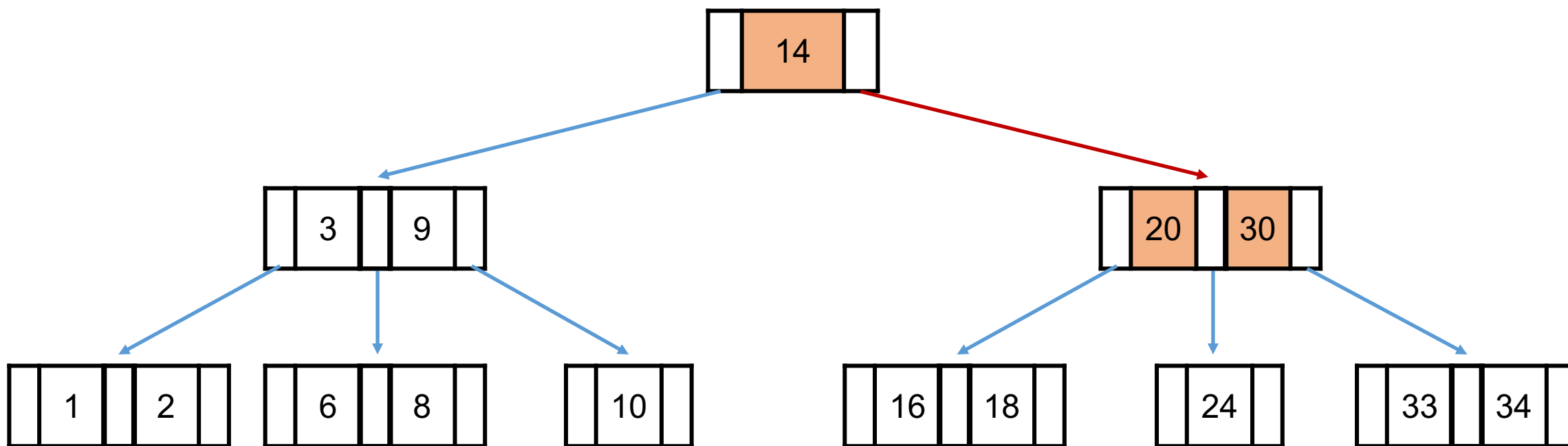
	26	
--	----	--



B 树 – 元素插入

插入:

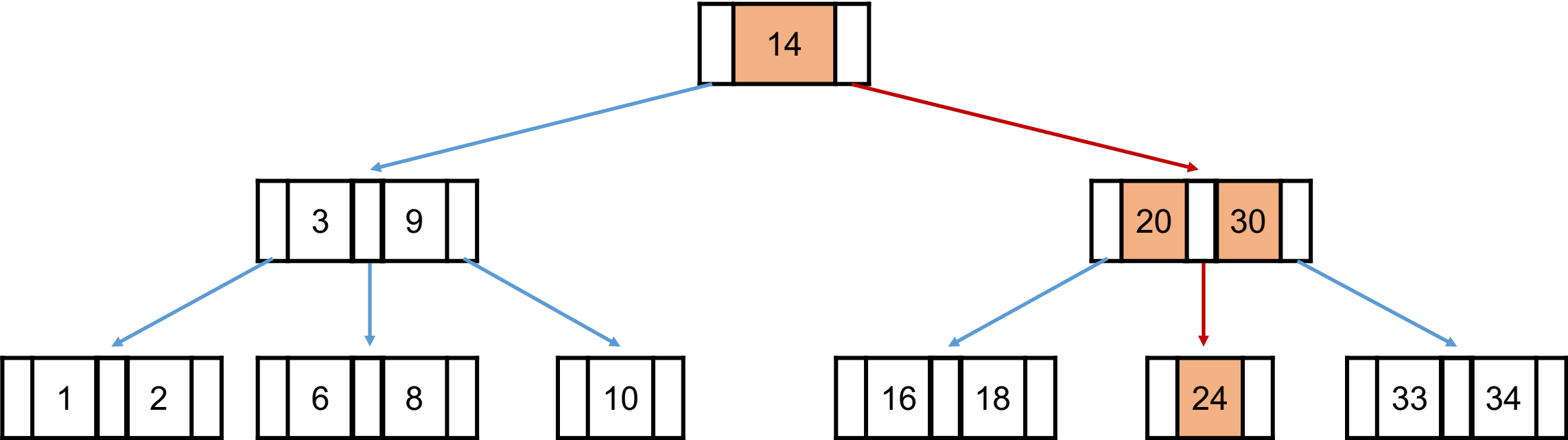
	26	
--	----	--



B 树 – 元素插入

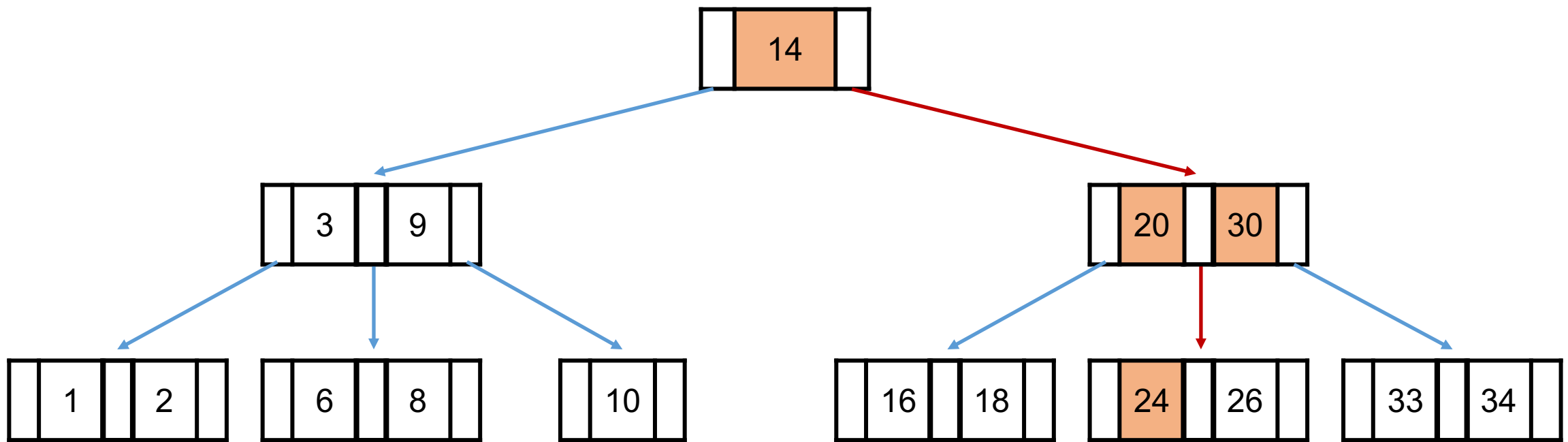
插入:

	26	
--	----	--

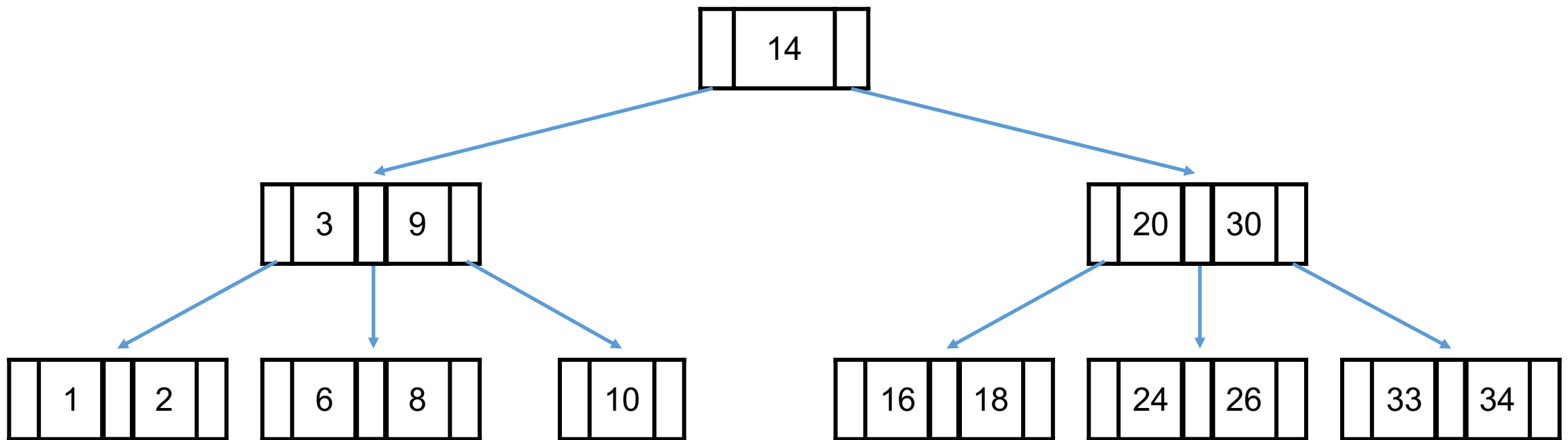


B 树 – 元素插入

插入:



B 树 – 元素插入





1. 元素插入

2. 插入调整

3. 元素删除

4. 删除调整





1. 元素插入

2. 插入调整

3. 元素删除

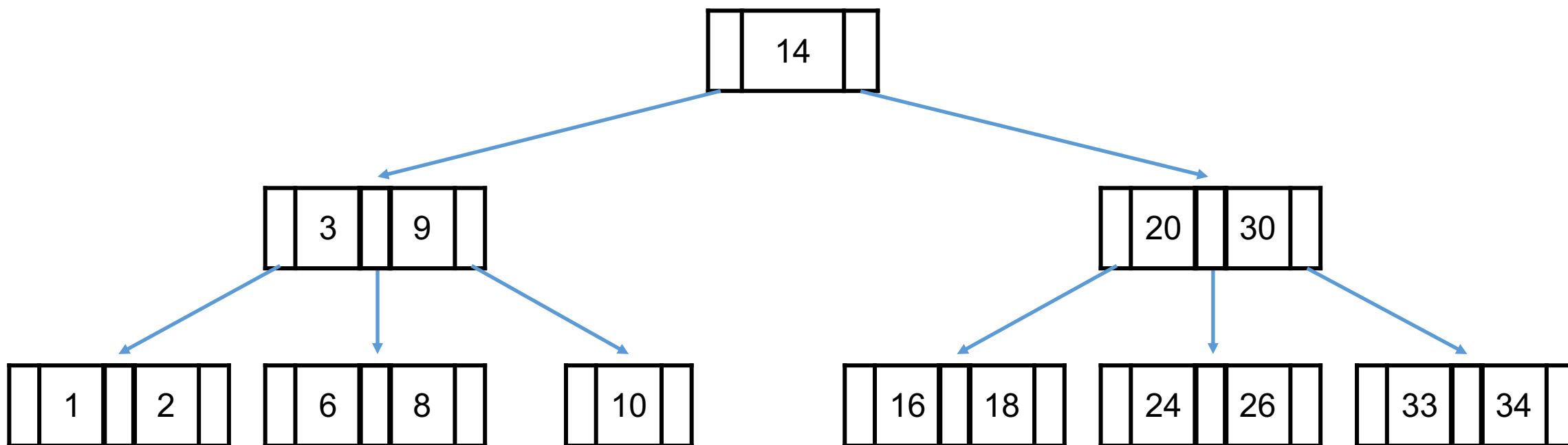
4. 删除调整



B 树 – 插入调整（上溢）

m 阶 B 树的插入调整：

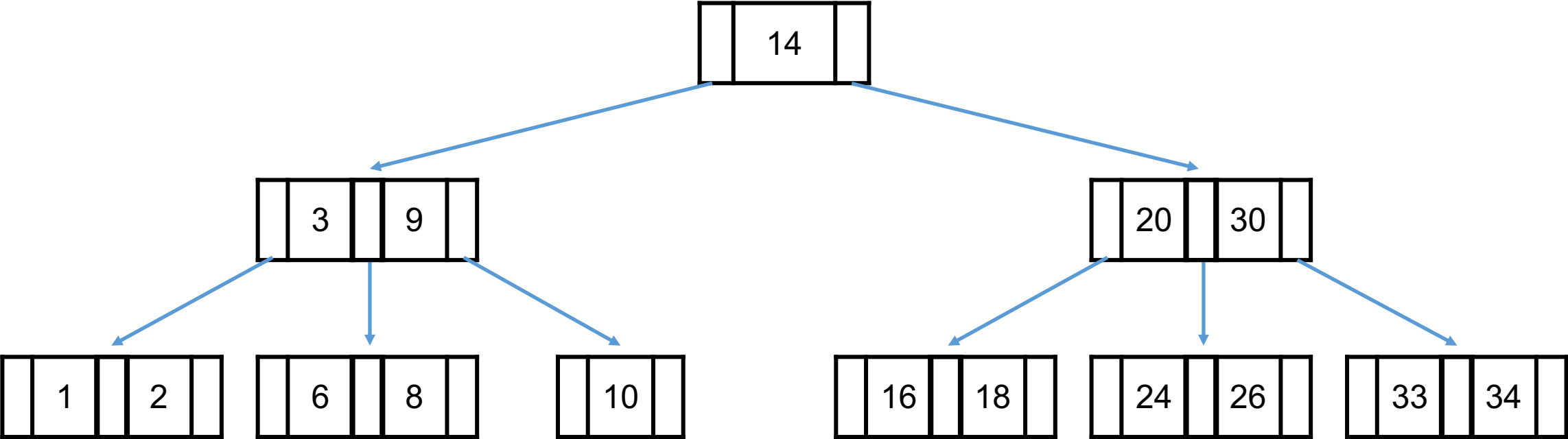
1. 插入调整站在父节点处理，发生在节点关键字数量达到 m 时
2. 插入调整的核心操作是：节点分裂



B 树 – 插入调整（上溢）

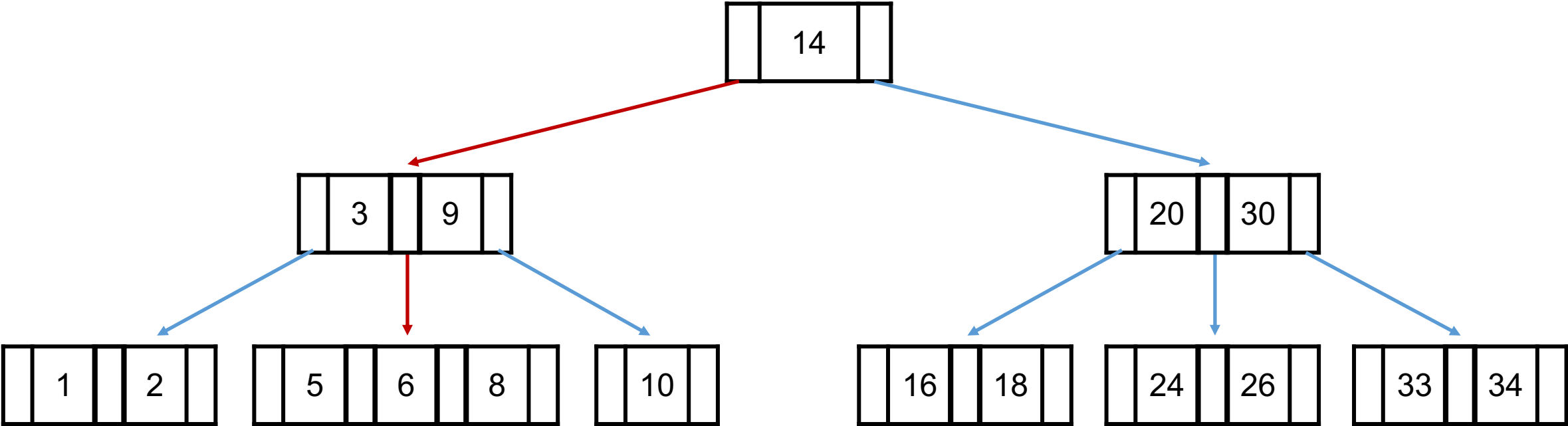
插入：

	5	
--	---	--



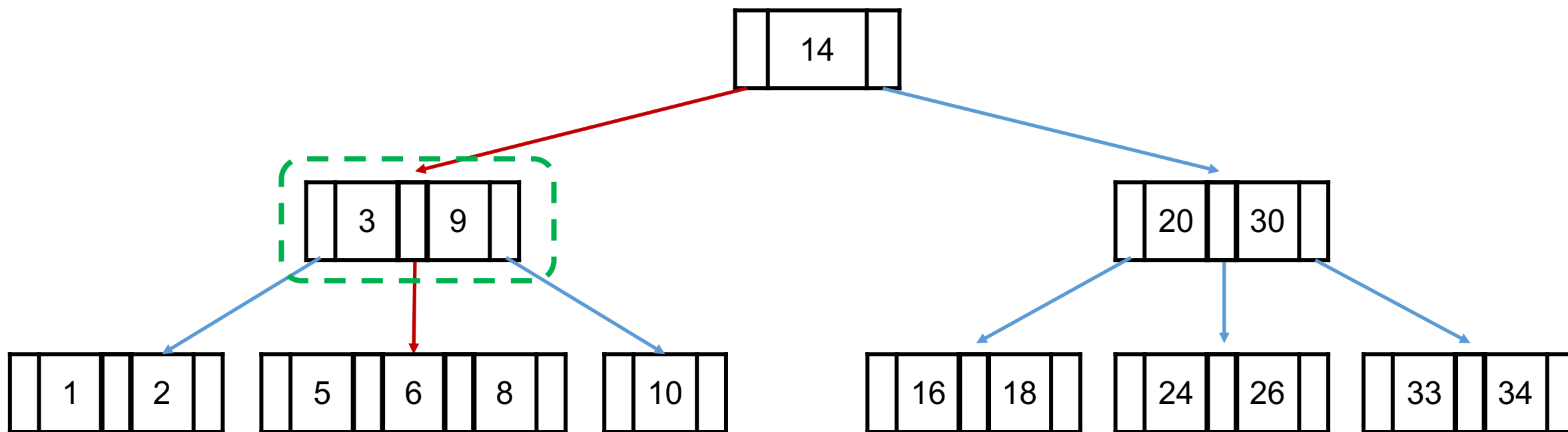
B 树 – 插入调整（上溢）

插入：



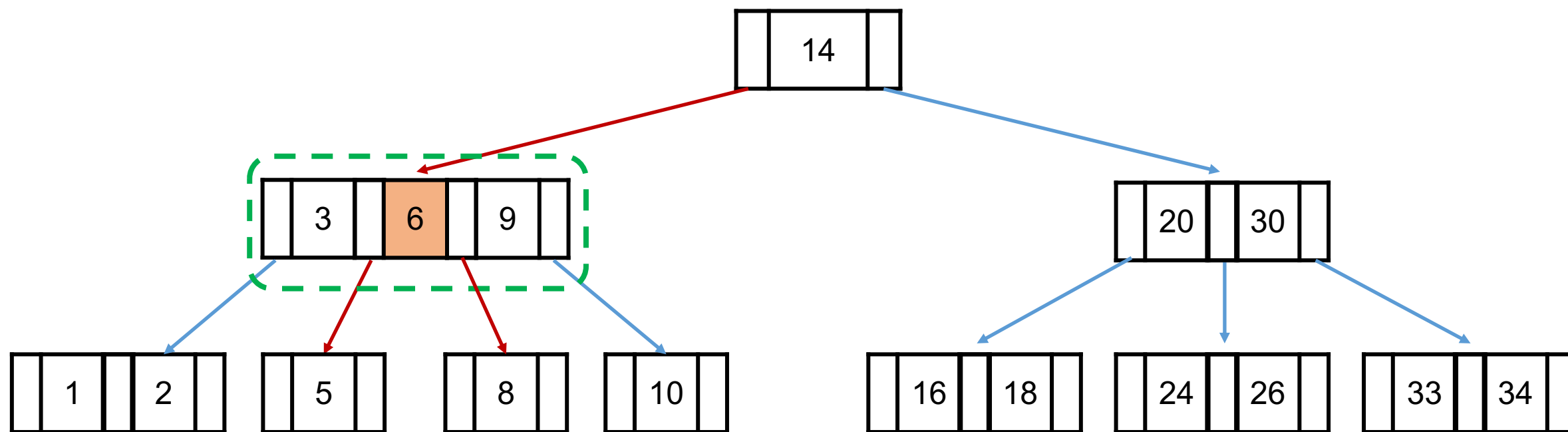
B 树 – 插入调整（上溢）

站在父节点处理



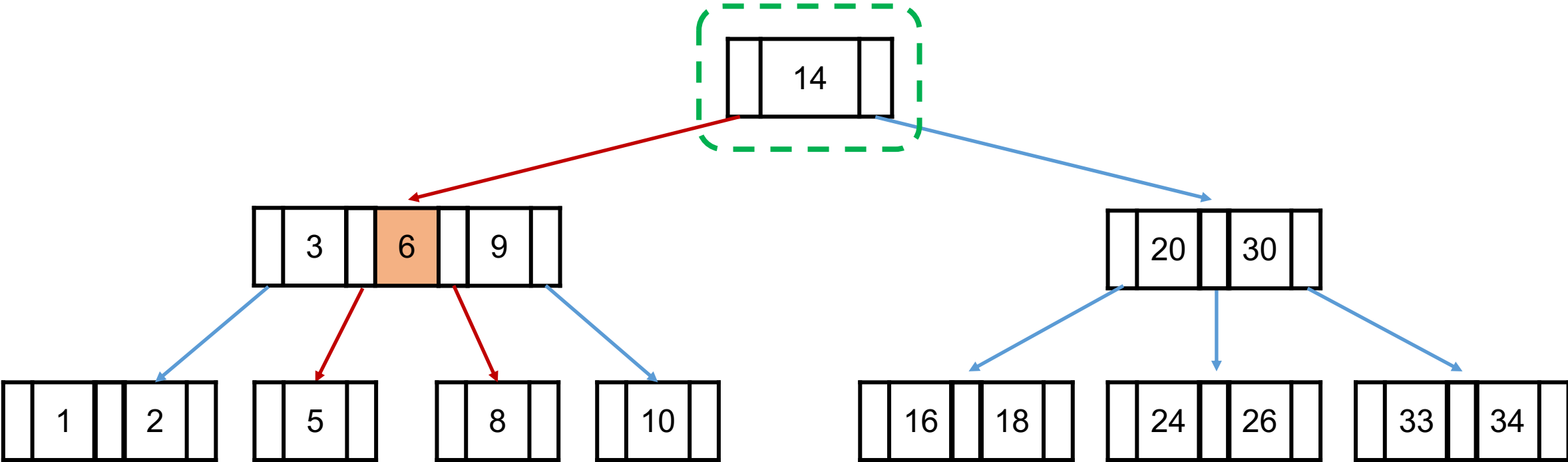
B 树 – 插入调整（上溢）

节点分裂



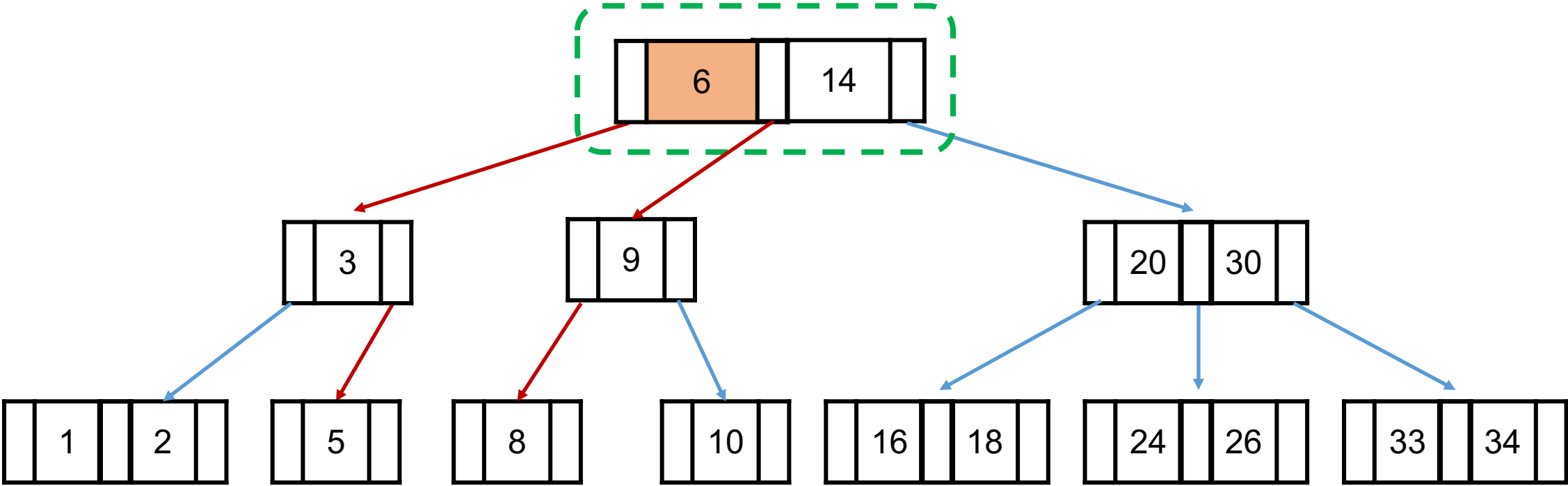
B 树 – 插入调整（上溢）

站在父节点处理

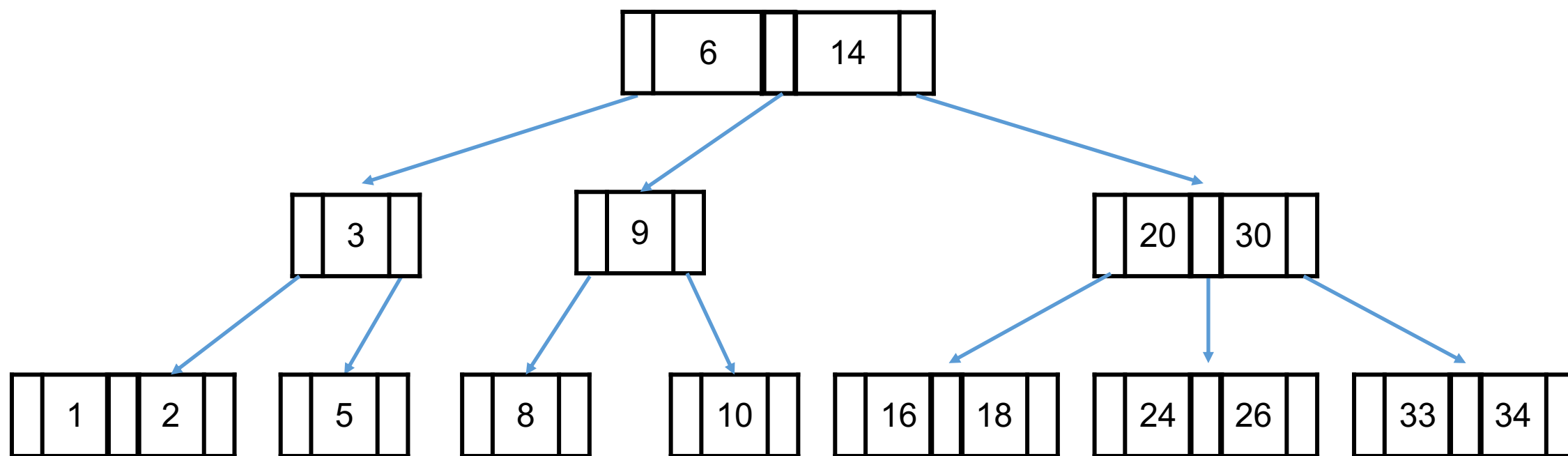


B 树 – 插入调整（上溢）

节点分裂



B 树 – 插入调整（上溢）





1. 元素插入

2. 插入调整

3. 元素删除

4. 删除调整



B 树 – 总结

m 阶 B 树，定义与操作总结：

1. 树中每个节点，最多含有 m 棵子树
2. 根节点中关键字数量范围 $1 \leq n \leq m$
3. 非根结点中关键字数量范围 $\lceil m/2 \rceil - 1 \leq n \leq m-1$
4. 插入调整为了解决『上溢』节点，关键字数量为 m 的节点
5. 删除调整为了解决『下溢』节点，关键字数量为 $\lceil m/2 \rceil - 2$ 的节点
6. 插入调整的核心操作是：节点分裂
7. 删除调整的核心操作是：左旋，右旋与合并