1 B+ Tree

- Insert 7 at the leaf node [6 8], the node is not full so we can add it and return

- Insert 9 at the leaf node [8 15], the node is full so we split it into nodes [8  15] and [9 15] and add 9 to the root node [6 8]. The root node is full so we split it into child nodes [6  8] and [9 15] and parent node [8  9].

- Insert 3 at the leaf node [0 1], the node is full so we split it into nodes [0  1] and [1 3] and add 1 to the root node [6  8]

- Insert 12 at the leaf node [9 15], the node is full so we split it into nodes [9  15] and [12 15] and add 12 to the root node [9  12]
• Remove 6 from the leaf node [6|7] and update the root accordingly, the node is at least half-full so we return

```
  8
 / \   /
1 7 9 12
 / \   /  \
0 1 3 7 8 9 12 15
```

• Remove 7 from the leaf node [7| ], the node is now less than half full so we borrow element 3 from sibling node [1|3] and update the root accordingly

```
  8
 / \
1 3 9 12
 / \
0 1 3 8 9 12 15
```

2 Dynamic Hash Index

```
  0       0 8 A
 /         / \
1 B
/  \
2 C
\  
3 D
```

• Insert 7 into bucket D pointed to by field 3 \equiv 7 \pmod{4}, the bucket is not full so we return

```
  0       0 8 A
 /         / \
1 B
/  \
2 C
\  
3 15 7 D
```

• Insert 9 into bucket B pointed to by field 1 \equiv 9 \pmod{4}, the bucket is not full so we return

```
  0       0 8 A
 /         / \
1 9 B
/  \
2 C
\  
3 15 7 D
```
• Insert 3 into bucket D pointed to by field 3 \equiv 3 \pmod{4}, the bucket is full so we split it. To add
the new bucket E we must grow the table's size to the next power of 2 which is 8. We set each new
field \(i\) to point to the old bucket at field \(i \mod 4\) except for 7 which points to the new bucket E. We
redistribute the elements of bucket \(D\), element 3 goes to bucket \(D\) pointed to by field \(3 \equiv 3 \pmod{8}\)
and elements 7 and 15 go to bucket E pointed to by field \(7 \equiv 15 \pmod{8}\).

• Insert 12 into bucket A pointed to by field 4 \equiv 12 \pmod{8}. This bucket is full so we split it, adding
new bucket F and setting field 4 to point to it. We redistribute the elements of bucket A. Elements 0
and 8 go to bucket A pointed to by field 0 \equiv 8 \pmod{8} and element 12 goes to bucket \(F\) pointed to
by field \(4 \equiv 12 \pmod{8}\).

• Remove 6 from bucket C. Bucket C is now empty so merge it into bucket A by updating pointers for
fields 6 and 2 to point to bucket A.
- Remove 7 from bucket E. The bucket is not empty so we return.