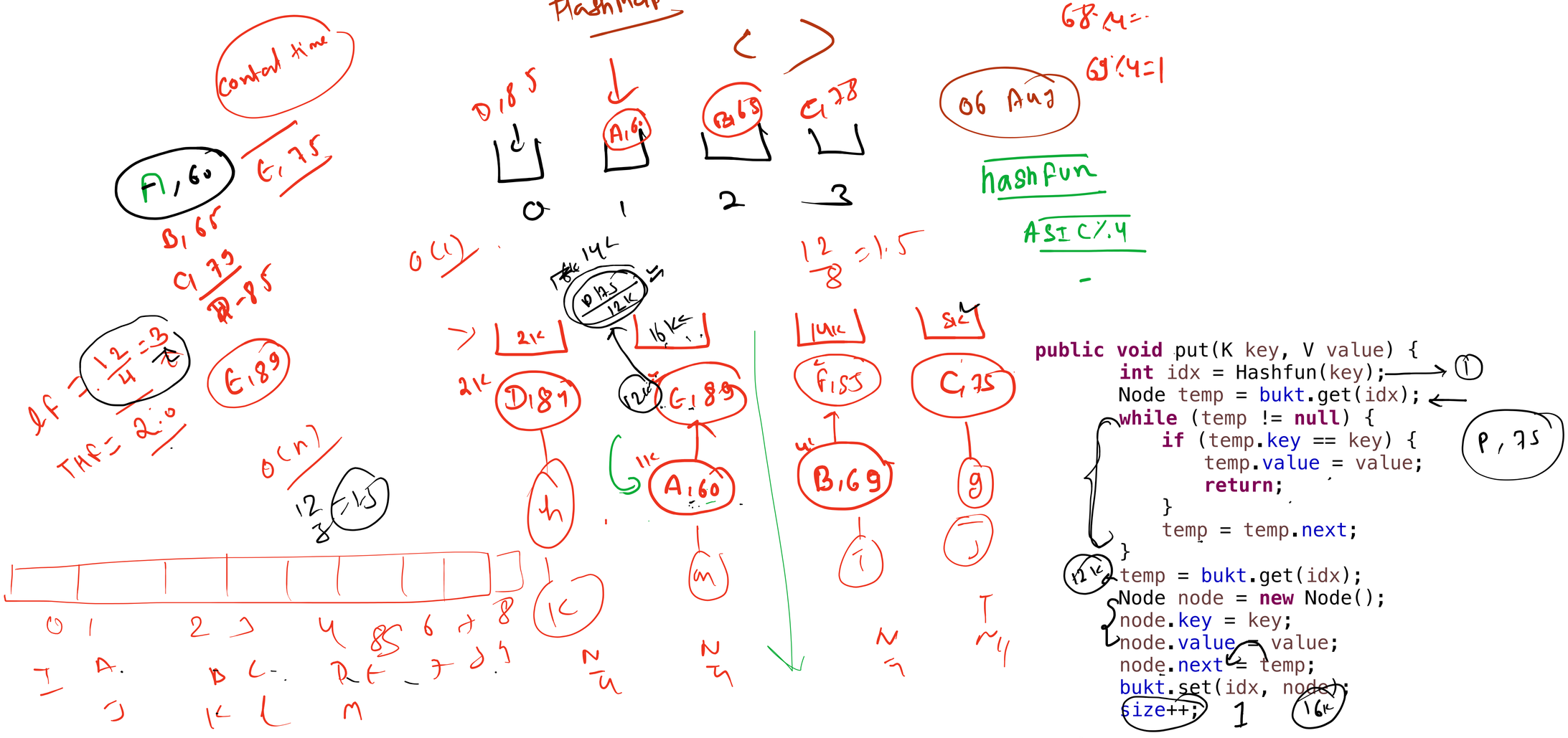
Lec-38

04/08/24 1:04 PM



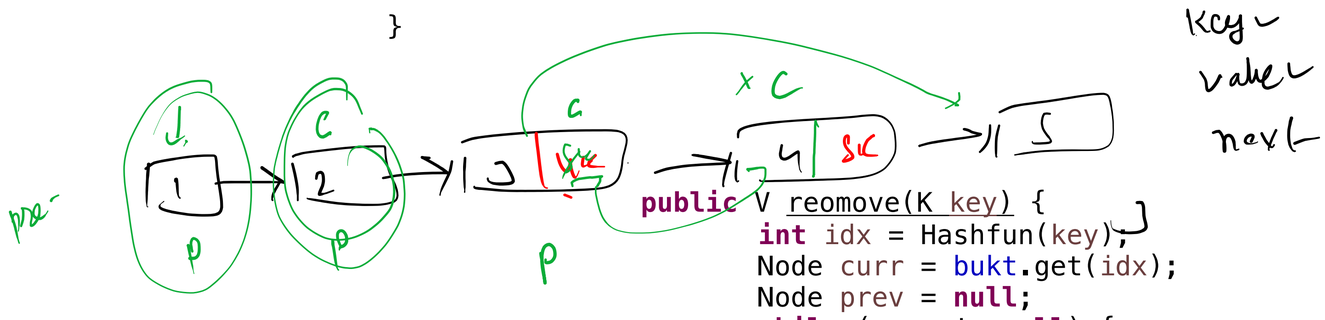

```
public void put(K key, V value) {
        int idx = Hashfun(key);
        Node temp = bukt.get(idx);
        while (temp != null) {
            if (temp.key == key) {
                temp.value = value;
                return;
            }
            temp = temp.next;
        }
        temp = bukt.get(idx);
        Node node = new Node();
        node.key = key;
        node.value = value;
        node.next = temp;
        bukt.set(idx, node);
        size++;
    }
}
```

P, 75

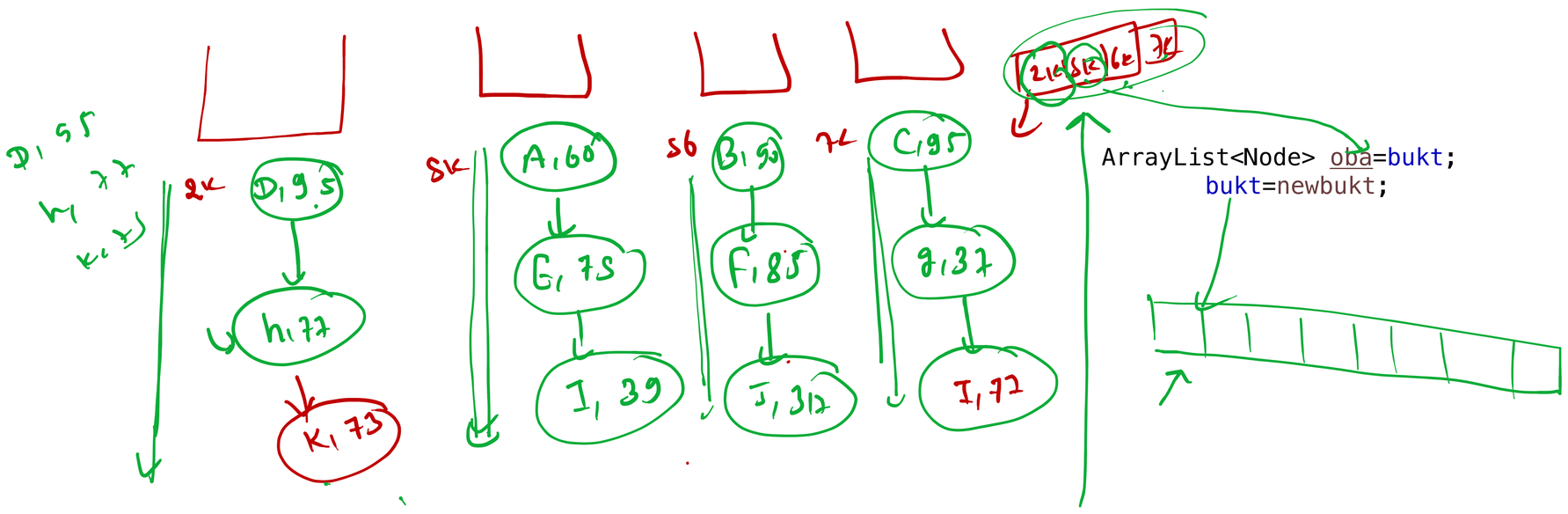
```
public void put(K key, V value) {

    }
```

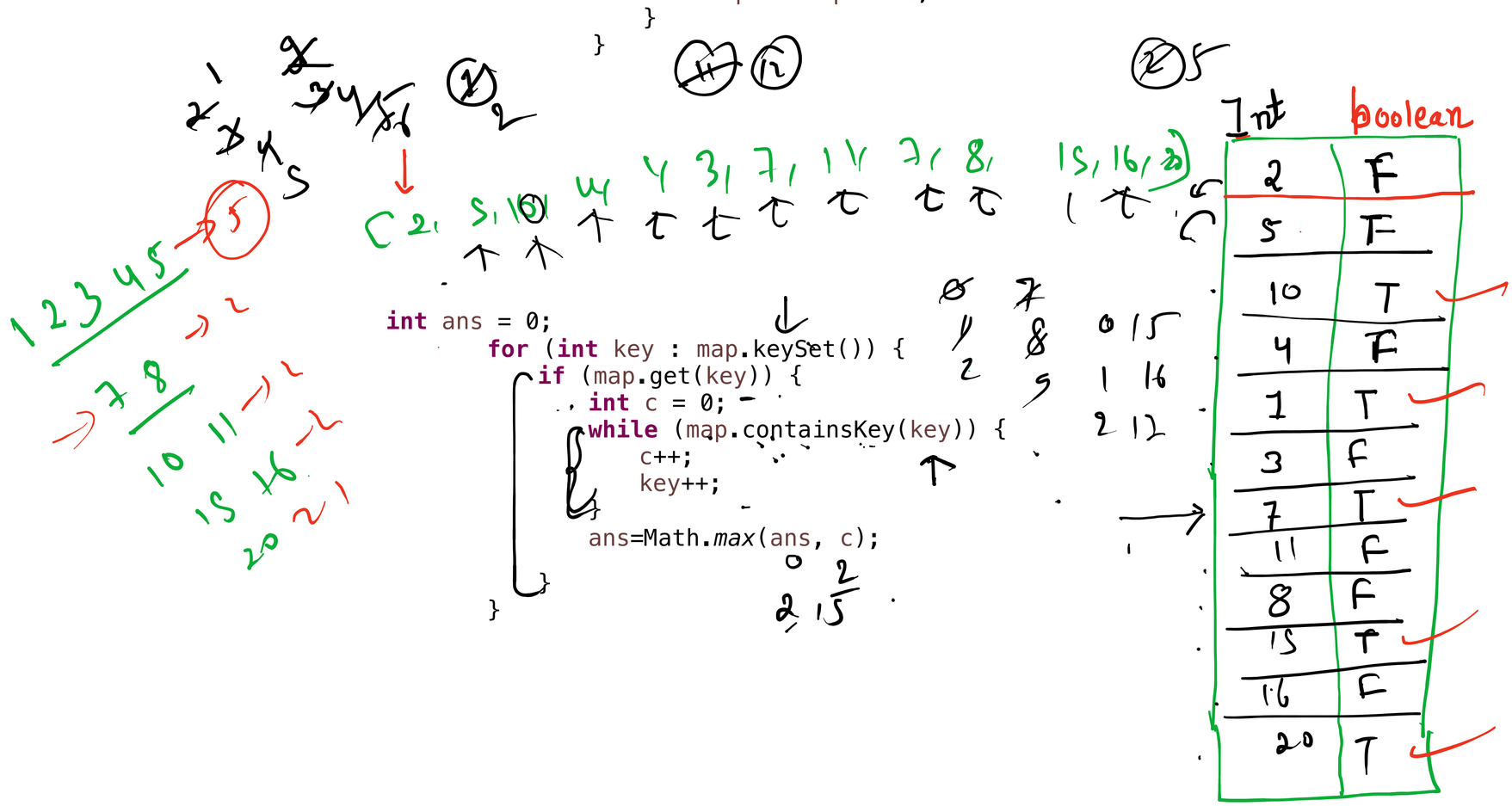
Class Node {
key
value
next




```
public V reomove(K key) {
    int idx = Hashfun(key);
    Node curr = bukt.get(idx);
    Node prev = null;
    while (curr != null) {
        if (curr.key.equals(key)) {
            break;
        }
        prev = curr;
        curr = curr.next;
    }
    if(curr==null) {
        return null;
    }
    else if(prev==null) {
        bukt.set(idx, curr.next);
        size--;
    }
    else {
        prev.next=curr.next;
        size--;
    }
}
```




```
for (Node temp : oba) {
            while (temp != null) {
                put(temp.key, temp.value);
                temp = temp.next;
            }
        }
```




```
int ans = 0;
        for (int key : map.keySet()) {
            if (map.get(key)) {
                int c = 0;
                while (map.containsKey(key)) {
                    c++;
                    key++;
                }
                ans=Math.max(ans, c);
            }
        }
```