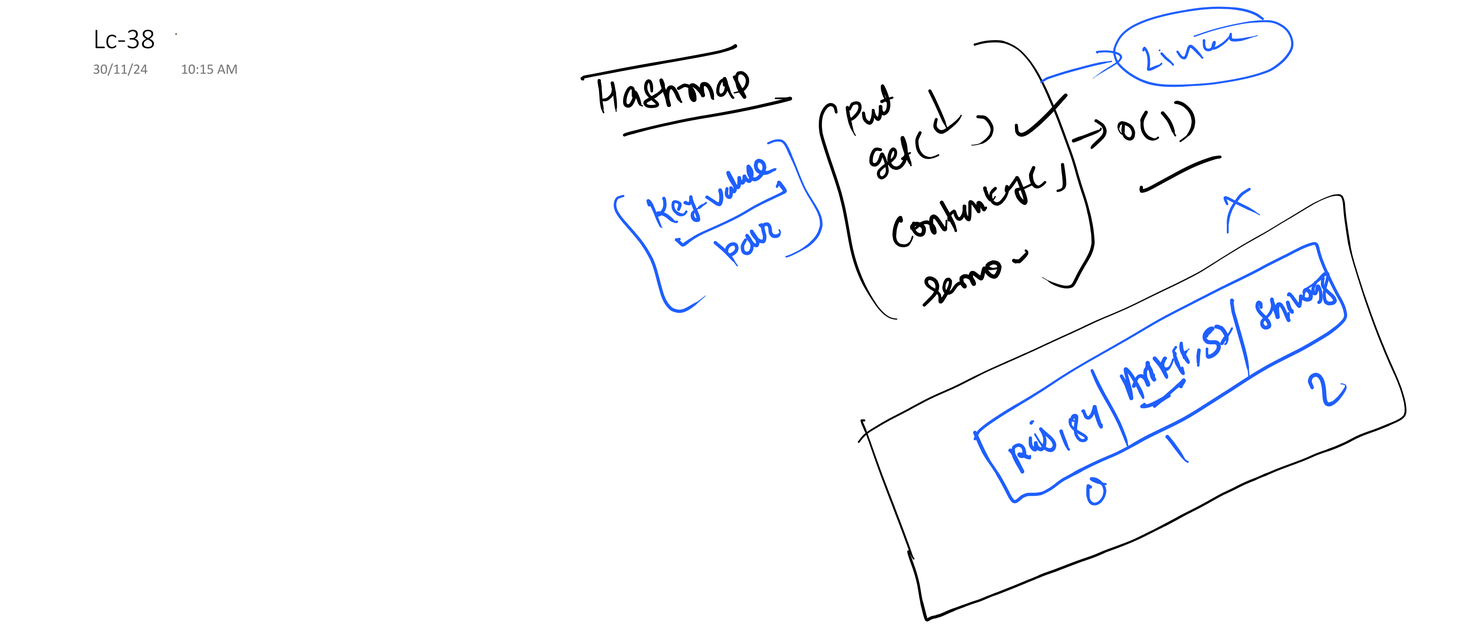
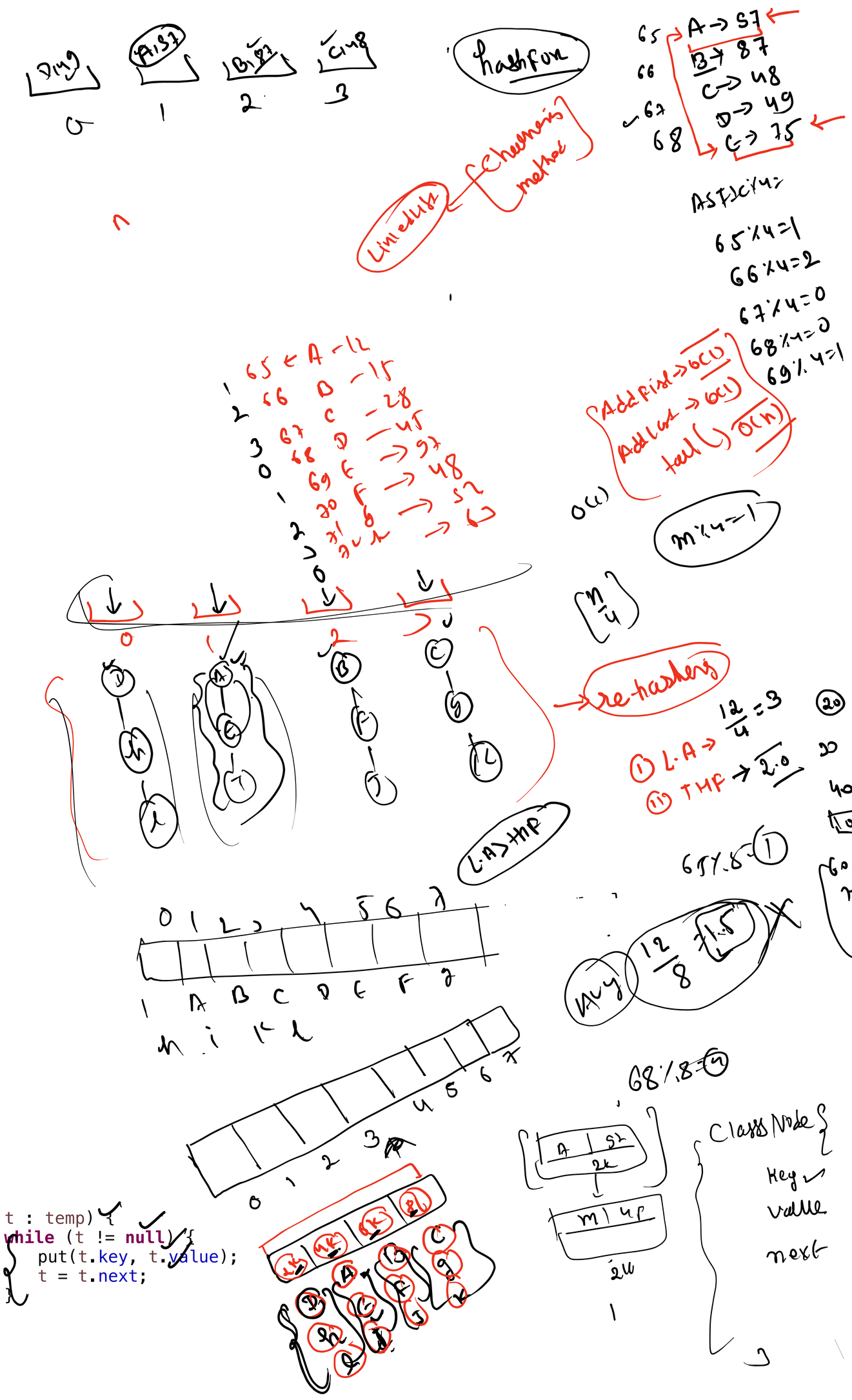
Lc-38

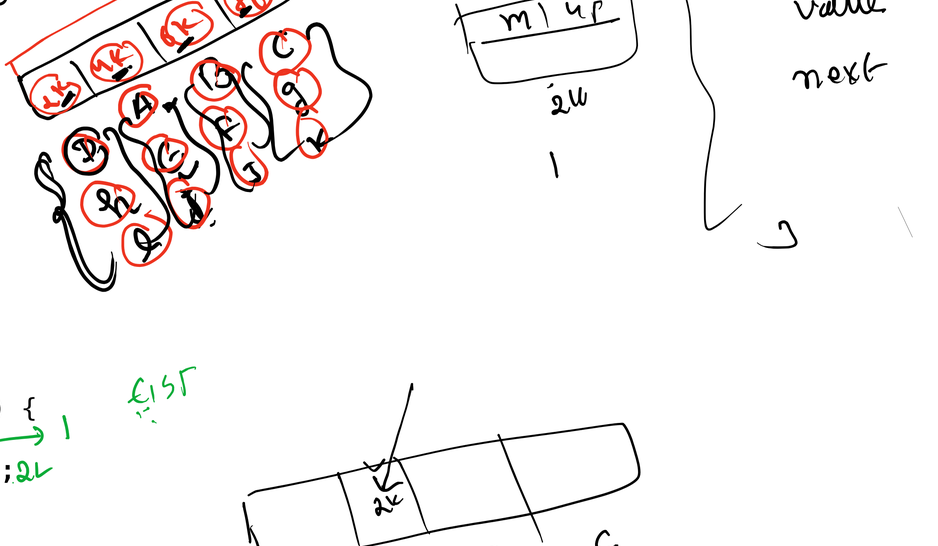
30/11/24 10:15 AM






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

```
    public void put(K key, V value) {
        int idx = hashfun(key);
        Node nn = buckt.get(idx);
        while (nn != null) {
            if (nn.key == key) {
                nn.value = value;
                return;
            }
            nn = nn.next;
        }
        Node n = new Node();
        n.key = key;
        n.value = value;
        nn = buckt.get(idx);
    }
```




```
public V remove(K key) {
    int idx = hashfun(key);
    Node curr = buckt.get(idx);
    Node prev = null;
    while (curr != null) {
        if (curr.key.equals(key)) {

        }
        prev = curr;
        curr = curr.next;
    }
}
```

```
    public V remove(K key) {
        int idx = hashfun(key);
        Node curr = buckt.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) {
            buckt.set(idx, curr.next);
        } else {
            prev.next = curr.next;
        }
        size--;
        return curr.value;
    }
```