

Hashfun  $\rightarrow$  Rabin Karp

Set/get  $\rightarrow$  array/ArrayList

HashMap

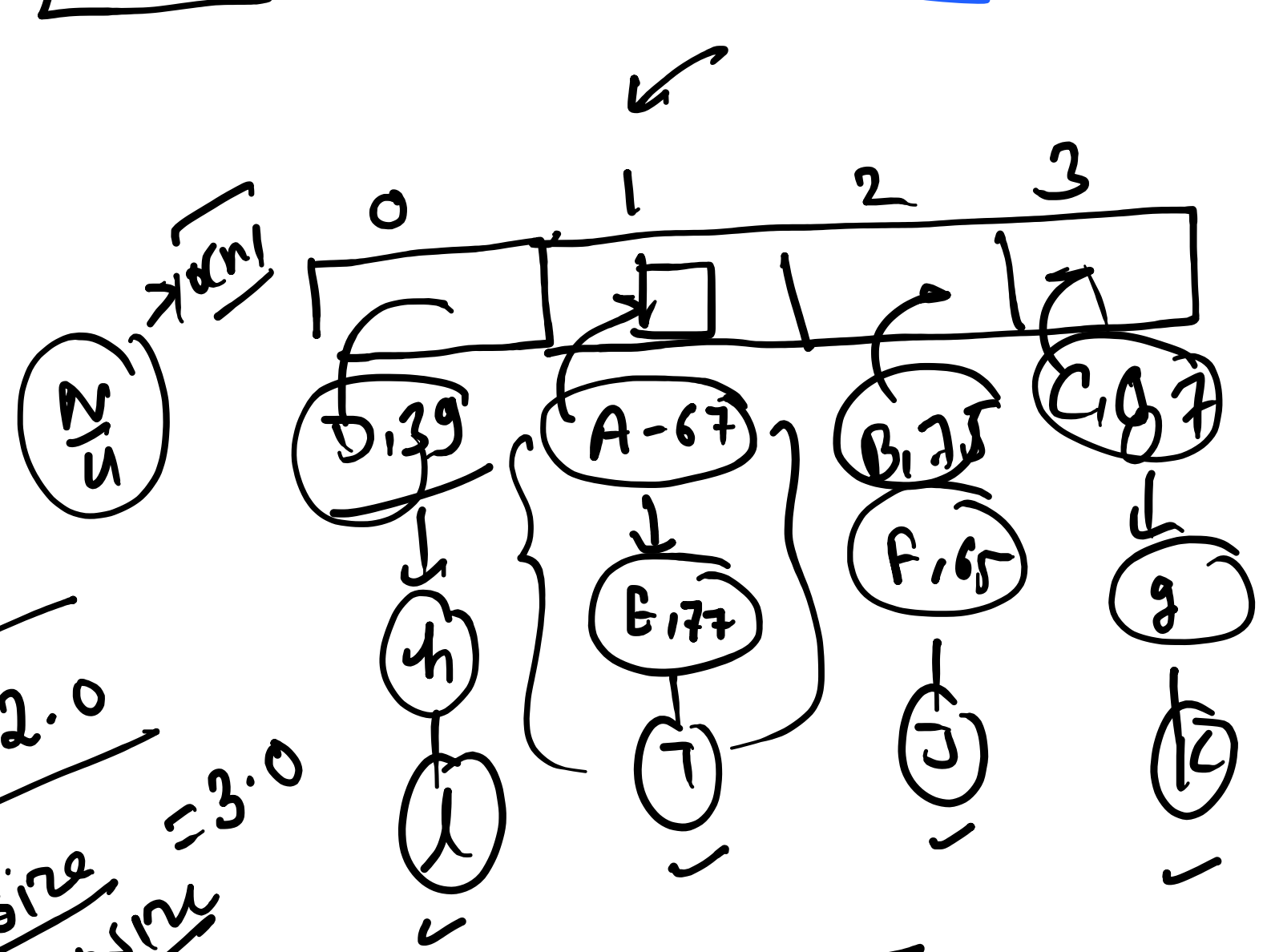
put/get/remove/contains

$65 \% 4 = 1$   
 $67 \% 4 = 3$   
 $68 \% 4 = 2$   
 $69 \% 4 = 0$

$THP = 2.0$   
 $IF = \frac{Size}{bukt.size} = 3.0$

$LA > THP$

rehashing



Key-value

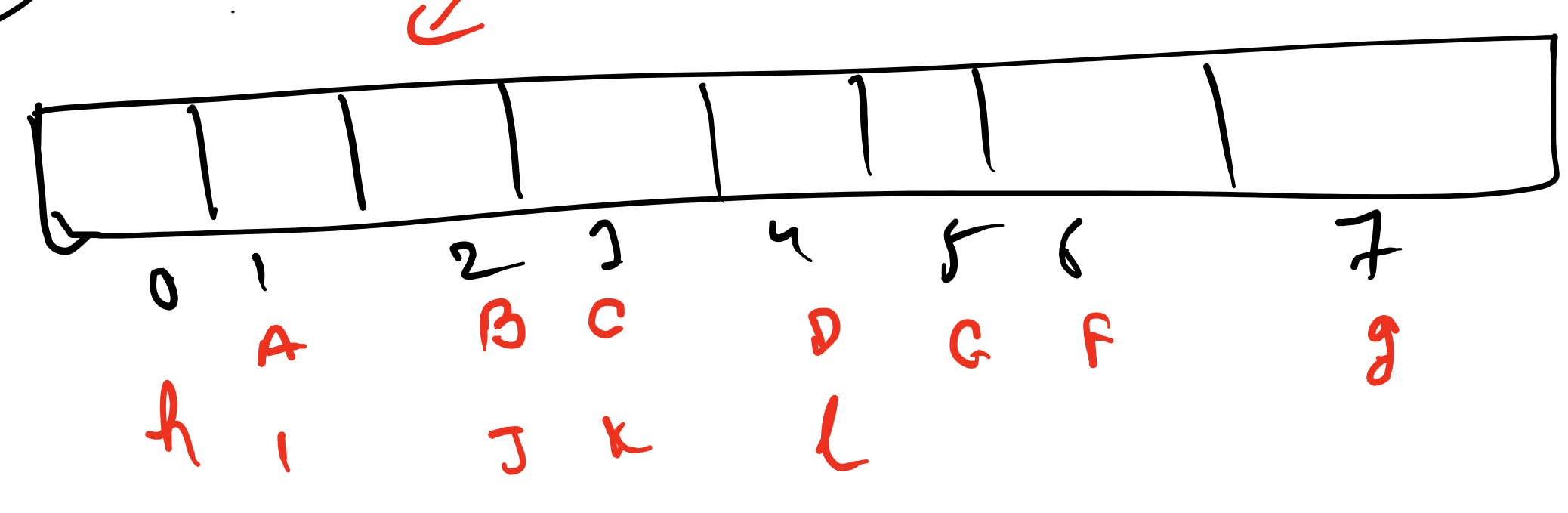
AS % 4

A  $\rightarrow$  76  
C  $\rightarrow$  87  
B  $\rightarrow$  75  
D  $\rightarrow$  39  
E  $\rightarrow$  77

get  
remove  
contains

$O(1)$

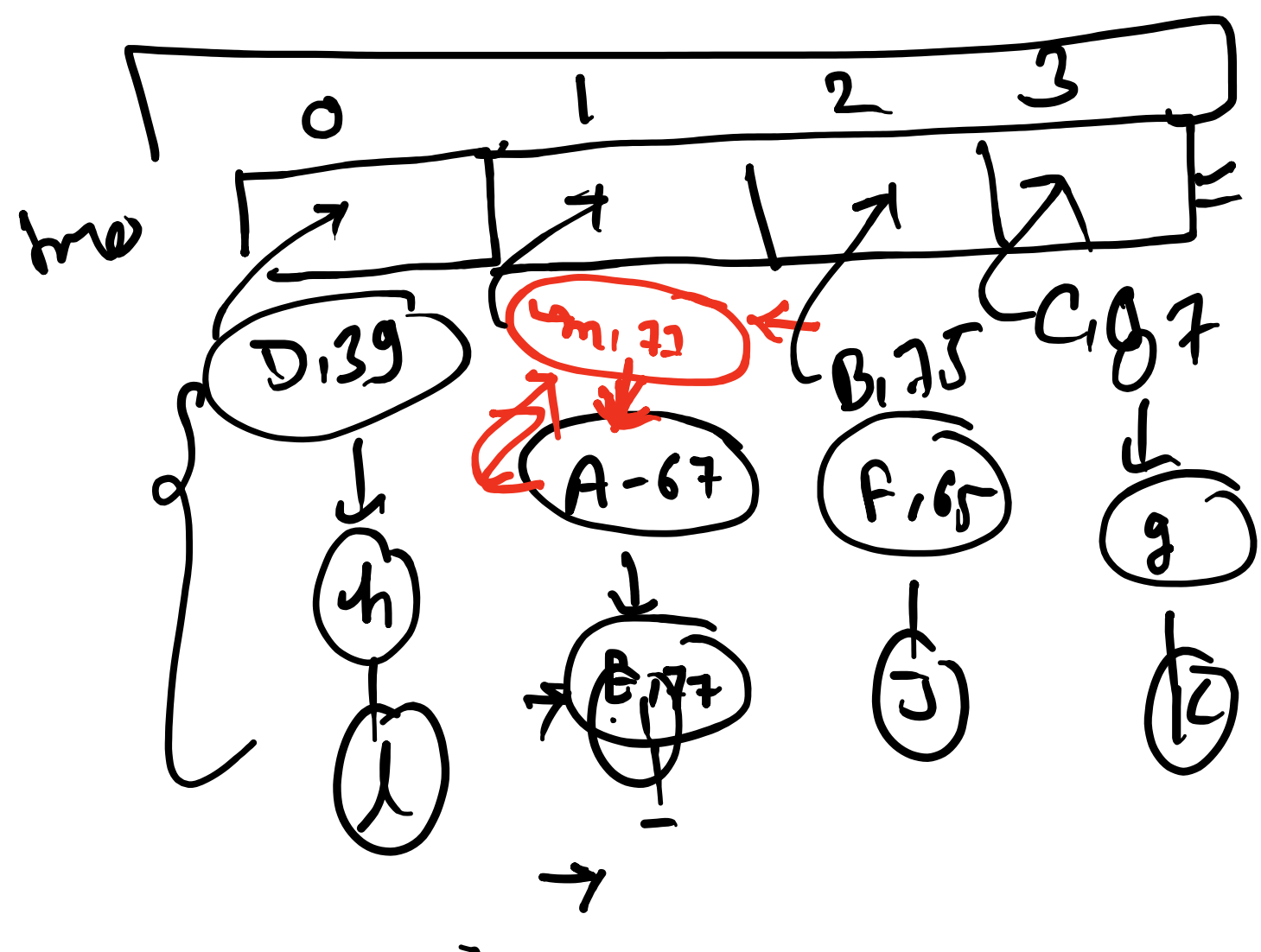
class Node {  
Key  
Value  
next



Ani KA  $\rightarrow$  I  
Ankita  $\rightarrow$  I

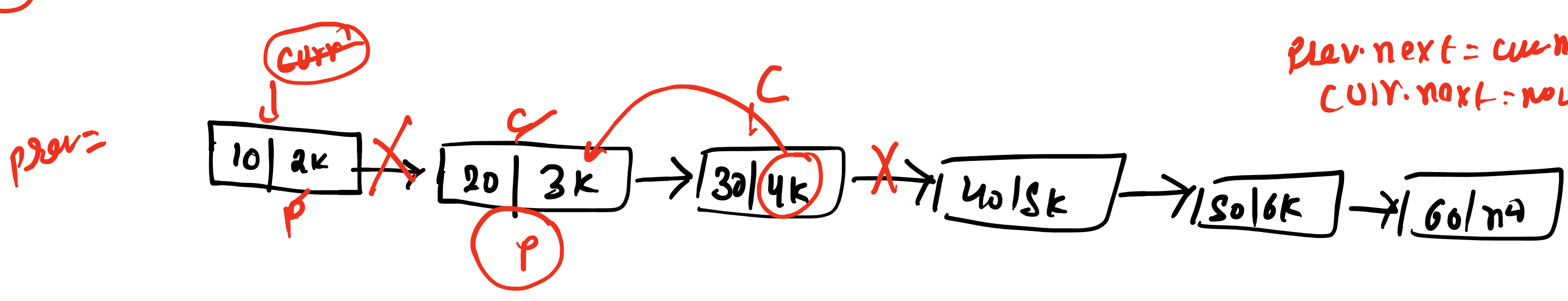
```
public void put(K key, V value) {  
    int idx = Hashfun(key);  
    Node temp = bukt.get(idx);  
    while (temp != null) {  
        if (temp.key.equals(key)) {  
            temp.value = value;  
            return;  
        }  
        temp = temp.next;  
    }  
    Node nn = new Node(); 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

```
ArrayList<Node> new_bukt = new ArrayList<>();  
for (int i = 0; i < 2*bukt.size(); i++) {  
    new_bukt.add(null);  
}  
ArrayList<Node> oba = bukt;  
bukt = new_bukt; size = 0
```



$m = 77$   
Key Value  
 $\frac{12}{4} = 3.0$

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



```
public V remove(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;  
    }  
}
```

$2, 2, 5, 10, 1$   
 $4, 1, 3, 7, 14, 3, 8, 1, 5, 16, 20$   
 $4, 8, 16, 20$   
 $16, 11$

1 2 3 4 5 = X 5  
7 8 X -2  
10 11 X -2  
18 X -2  
20 X -1  
Key = 7, 8, 9

c = 0 | 2  
while (set.contains(key)) {  
 set.remove(key);  
 c++;  
 key++;  
}  
ans = max(c, ans)