

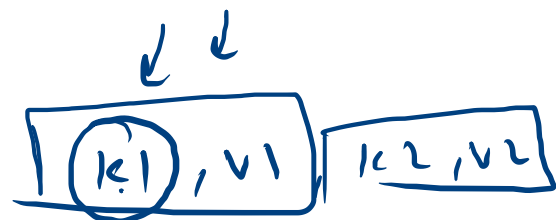
✓
α

put	get	ck	remove	keyset	size
key value	value null	True false	value 2 null	k1...kn	sn
$O(1)$	$O(1)$	$O(1)$	$O(1)$	~	$O(1)$

requirement

k	v
<u>k1</u>	<u>v1</u>
k2	v2
k3	v3

put



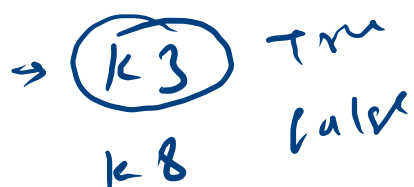
get



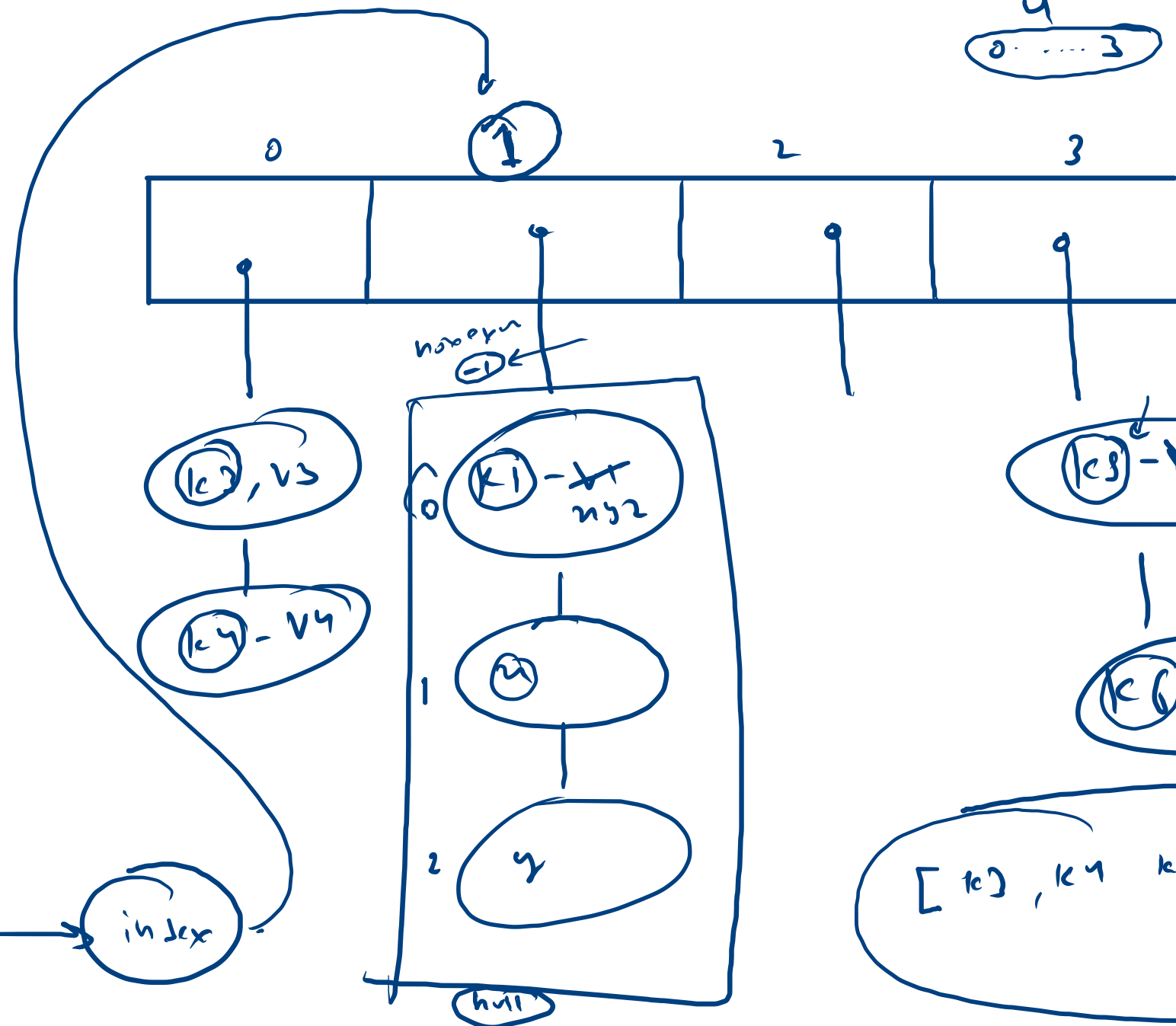
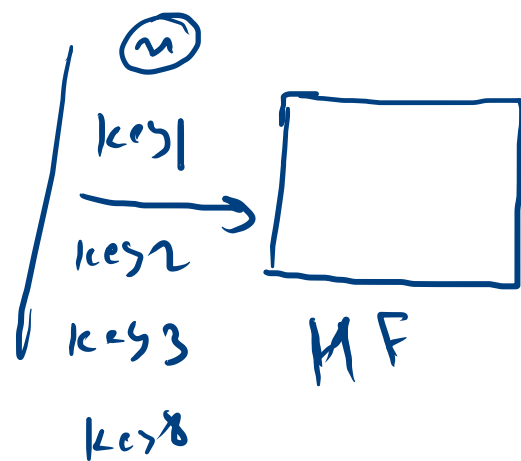
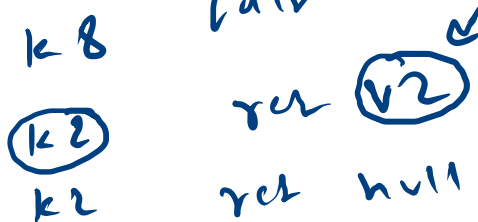
put



clear



remove

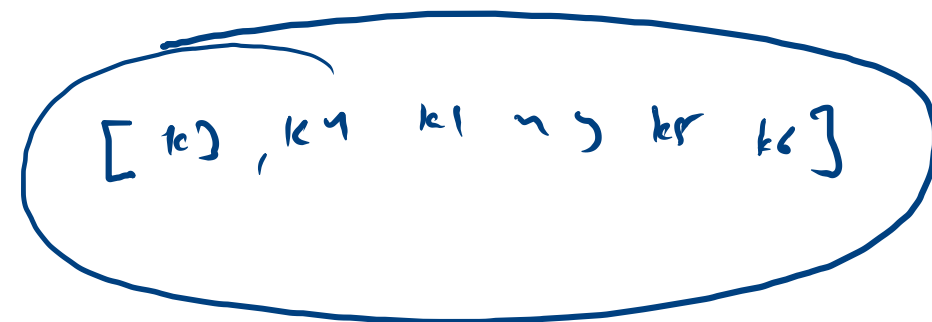
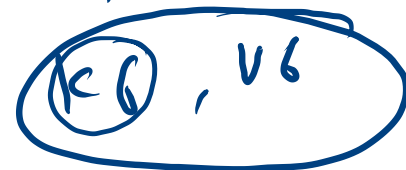
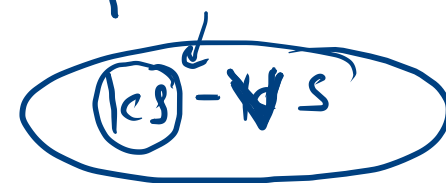


$N \rightarrow$ size of bucket

$h \rightarrow$ count of k-v

$$\lambda = \frac{h}{N} \frac{s}{n}$$

$$\lambda \leq \frac{k}{2}$$



100554

$$h \rightarrow \frac{9}{4}$$
$$N \rightarrow 4$$
$$k \rightarrow 2$$

$$\lambda \leq k$$
$$\lambda = \frac{9}{4} = 2.25$$
$$\frac{h}{N} \leq 1 \leq 2$$

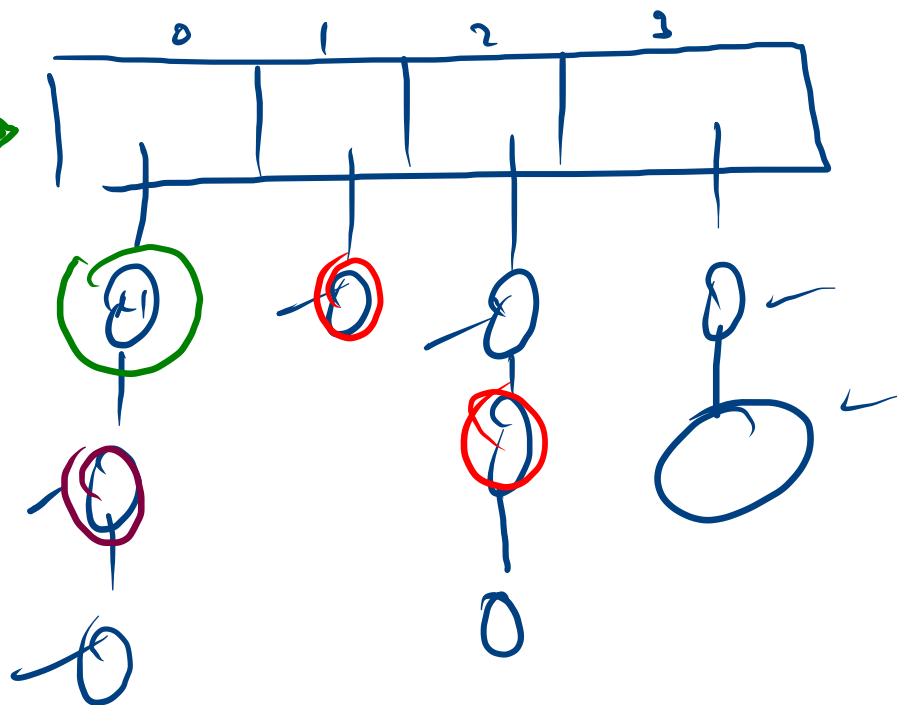
$$\frac{h}{N} \leq 2$$

$$\frac{h}{N} > 2 \rightarrow \text{rehash}$$
$$\text{size} \rightarrow h > 2 * N \rightarrow \text{b.l.r.}$$

new size 0

bucket

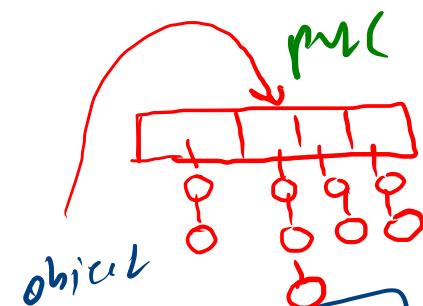
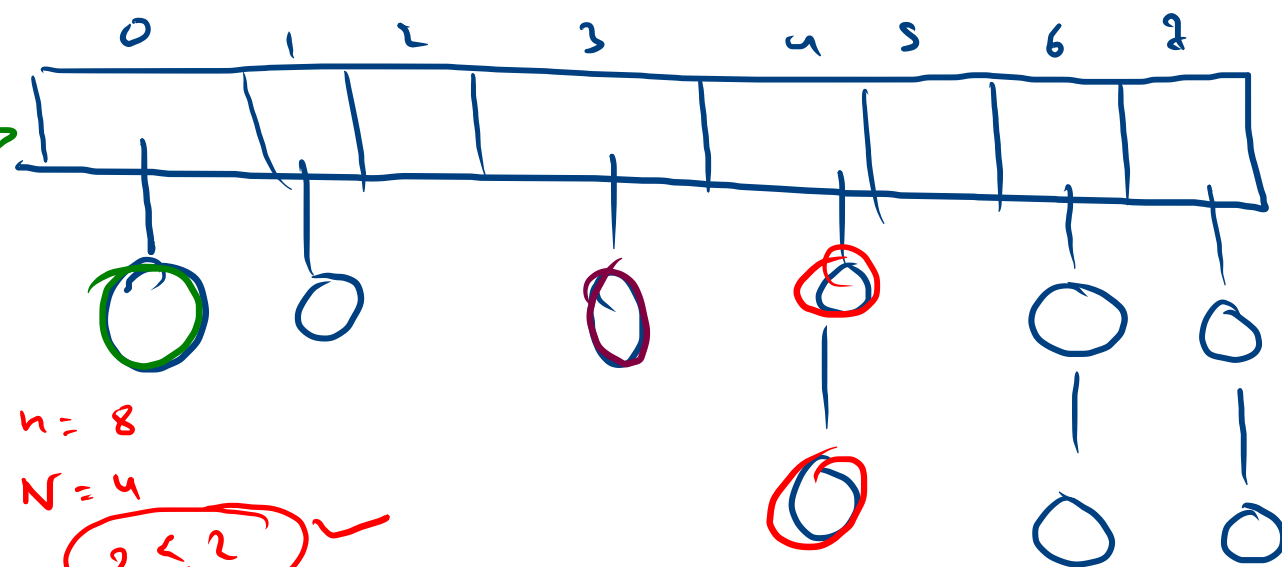
old bucket



re hashing

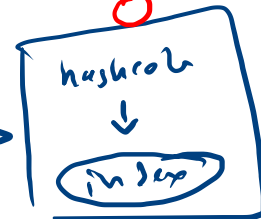
new size 0

bucket



object

key



index

$$h = 8$$

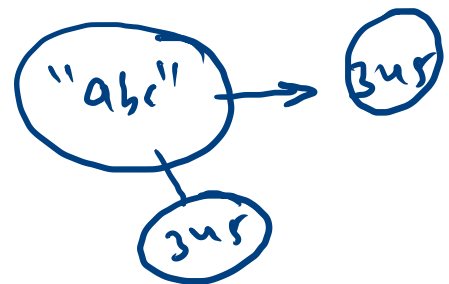
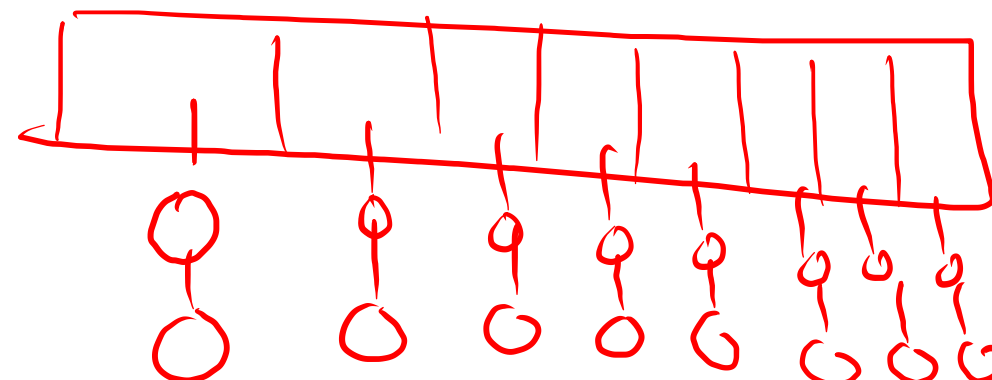
$$N = 4$$

$$2 \leq 2$$

$O(1)$ ✓

size 20

bucket



hashcode



$$\frac{h}{N} = 2 \leq 2$$

$$h = 16$$
$$N = 8$$

```
private class HMNode {
    K key;
    V value;

    HMNode(K key, V value) {
        this.key = key;
        this.value = value;
    }
}
```

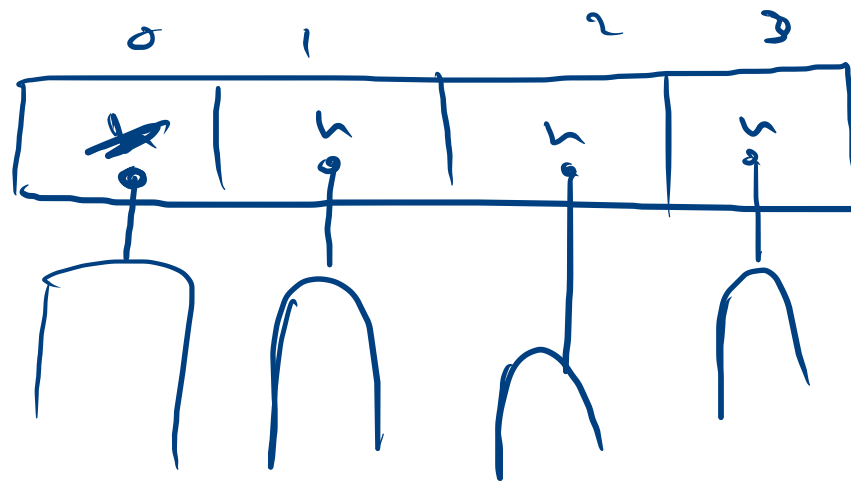
```
private int size; // n
private LinkedList<HMNode>[] buckets; // N = buckets
```

```
public HashMap() {
    initbuckets(4);
    size = 0;
}
```

```
private void initbuckets(int N) {
    buckets = new LinkedList[N];
    for (int bi = 0; bi < buckets.length; bi++) {
        buckets[bi] = new LinkedList<>();
    }
}
```

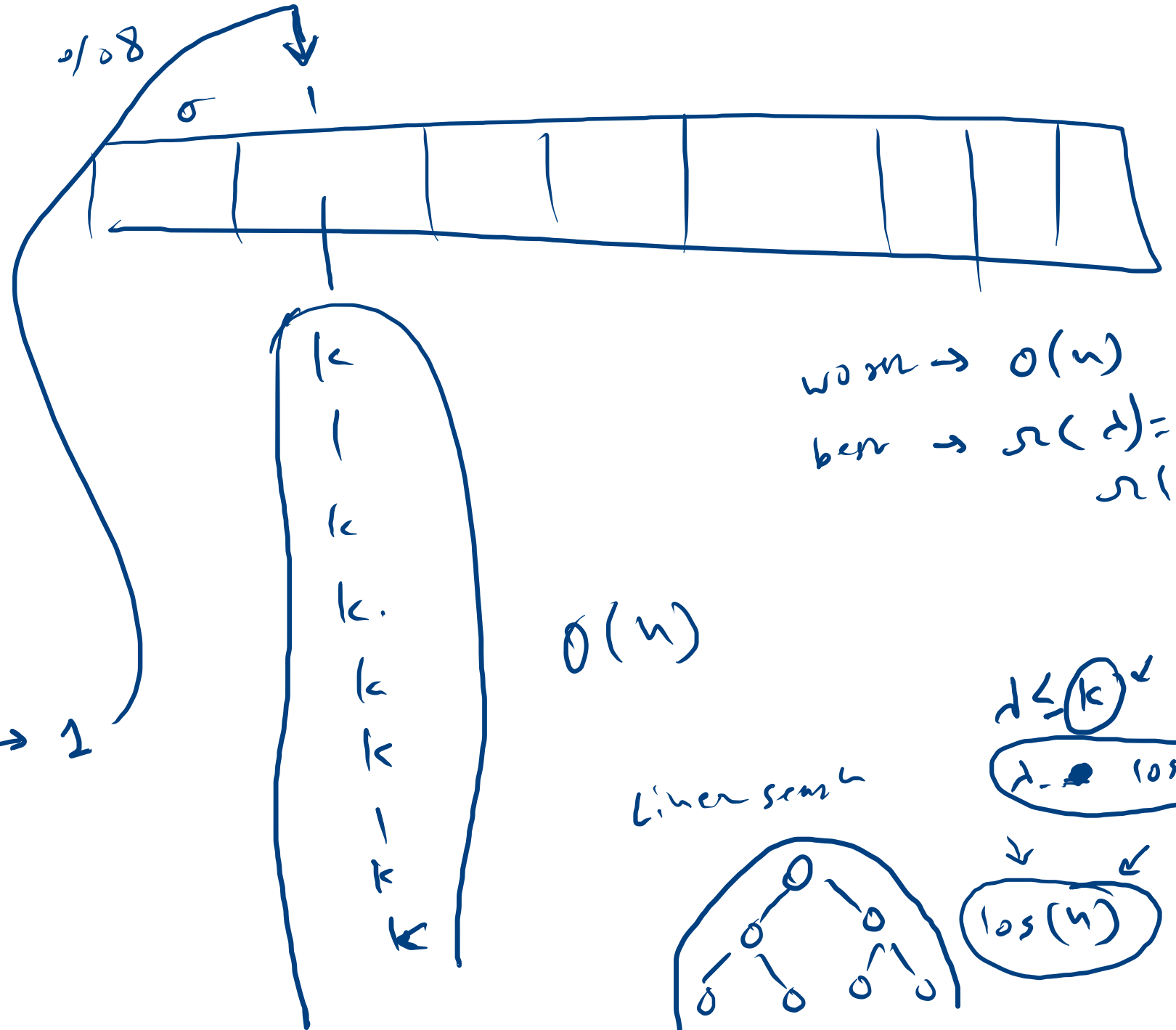
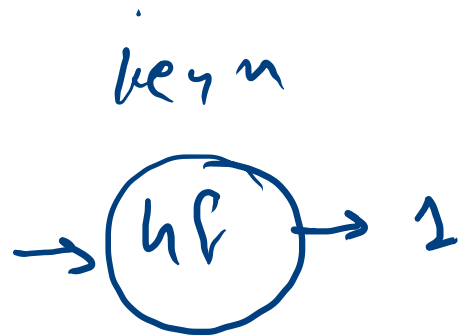
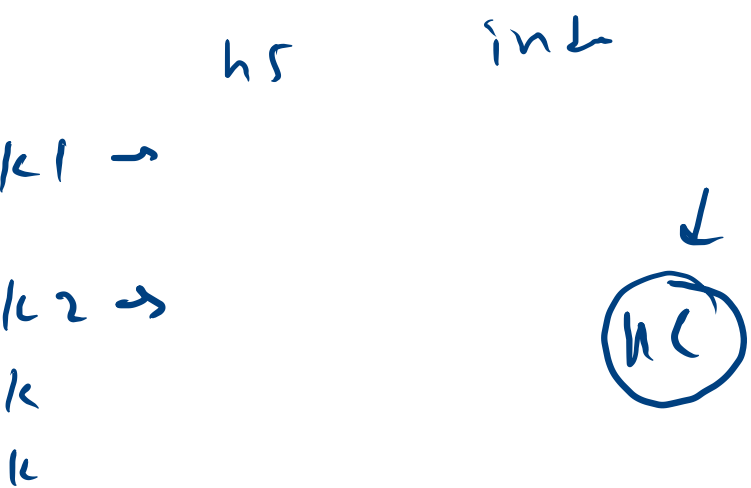
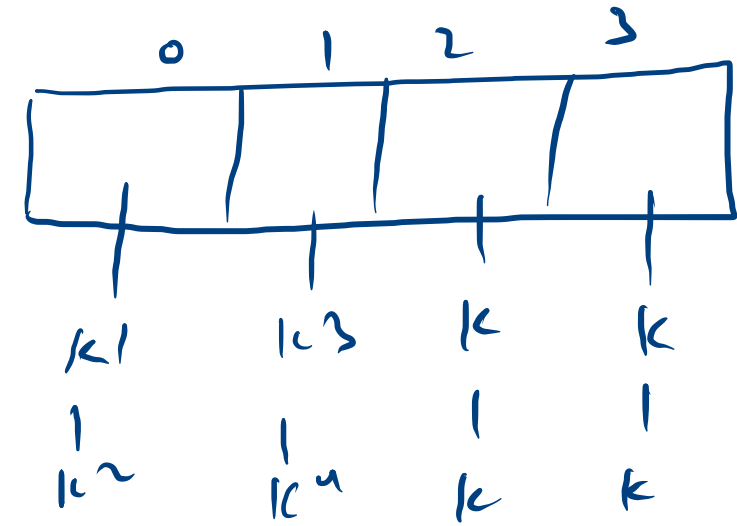
key
value

```
int a[] = new int[10000];
LinkedList b = new LinkedList();
```



hull

pu
all



worst $\rightarrow O(n)$
 best $\rightarrow \Omega(d) = \Omega(1)$

$O(n)$

Linear search

