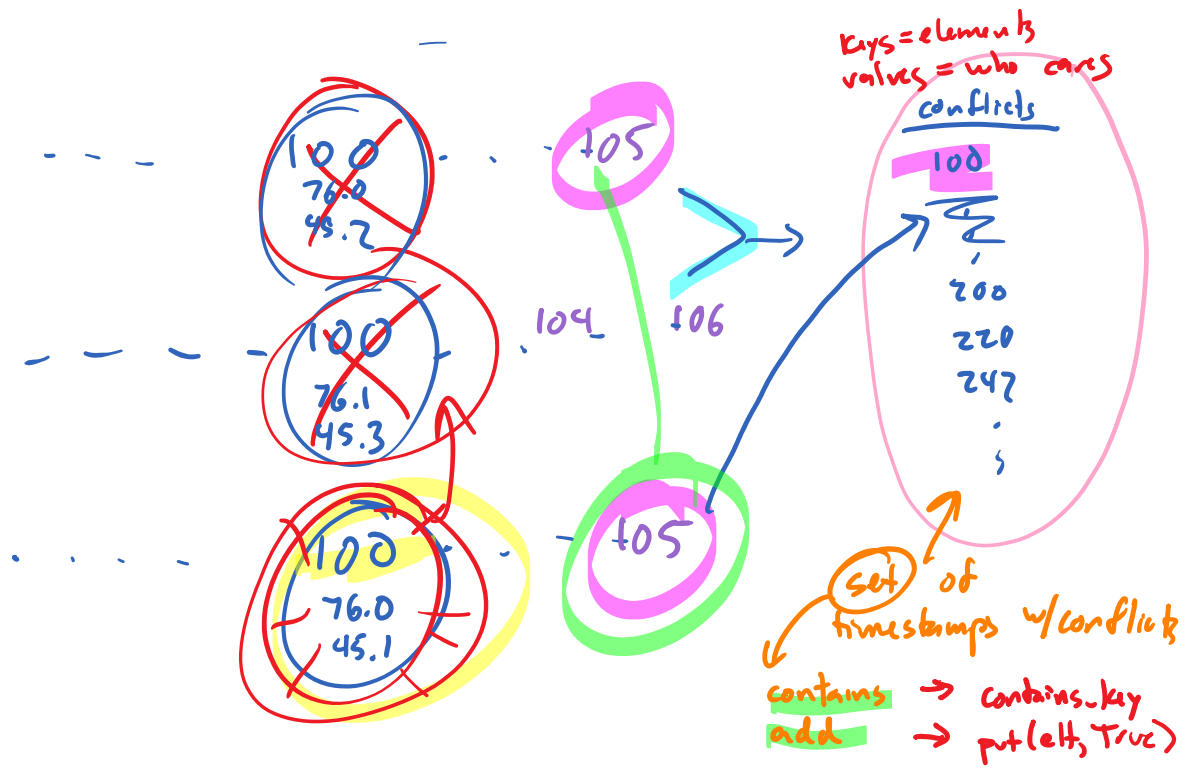


Set = Map w/ no values



```

20144987 4 4 4 CLT AVL X 17 1.00 X
20144214 3 4 4 IAH CLT - US 1.00 X
201441110 4 4 4 PHX OAK X US 1.00 X
20144794 3 4 4 CLE CLT - 16 1.00 X
20144756 2 4 4 CLT BOS X US 1.00 X
201441020 2 4 4 CLT MCO X US 1.00 X
201441578 4 5 4 BNA PHL - YX 1.00 X
201442030 4 4 4 DCA BHM X US 1.00 X
201442094 3 4 4 CHS CLT - 16 1.00 X
201441020 1 4 4 AVL CLT - 16 1.00 X
201441020 3 4 4 MCO CLT - US 1.00 X
    
```

hash table

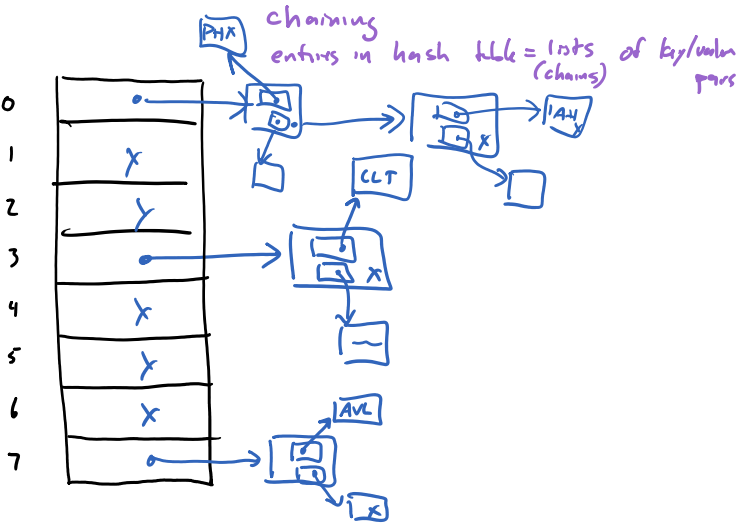
	key	value
0	IAH	1
1		
2		
3	CLT	1
4		
5		
6		
7	AVL	1

collision!

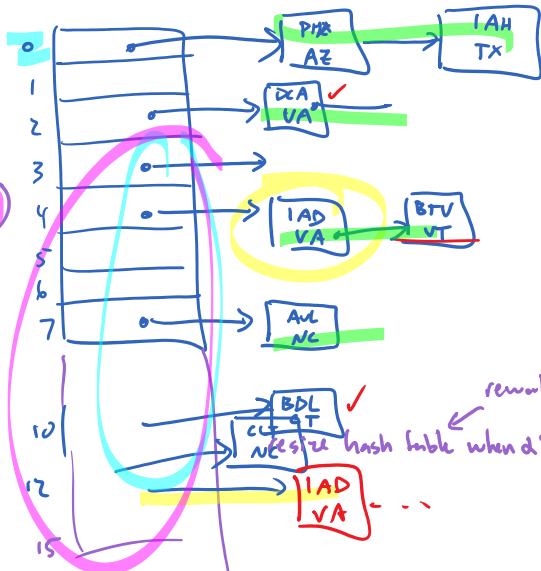
hash function

CLT → 66827 % 8 = 3
 AVL → 65207 % 8 = 7
 IAH → 72240 % 8 = 0
 PHX → 79200 % 8 = 0
 OAK
 CLE
 BOS
 MCO

hash fun
 CLT → 3
 AVL → 7
 IAH → 0
 PHX → 0
 OAK



chaining entries in hash table = lots of key/value pairs (chains)



put/get/contains
 - 1) $\text{hash}(\text{key}) \% m$ to find index of chain
 2) search chain to find key
 3) ...
 get(IAD)

hash table w/ m slots (chains)

n entries (key/value pairs)

each chain has length

$$\alpha = \frac{n}{m}$$

load factor

so $m \geq n$ hence $\alpha \leq 1$

so expected $O(1)$ time for get/put/contains_key

(still worst-case $O(n)$)

