

```

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

hash table

key	value
0	
1	OAK 1
2	
3	CLT 2
4	
5	
6	BOS 1
7	X

CLT 65827 % 8 = 3  
 OAK 78009 % 8 = 1  
 BOS 65958 % 8 = 6  
 MCO 76153 % 8 = 1

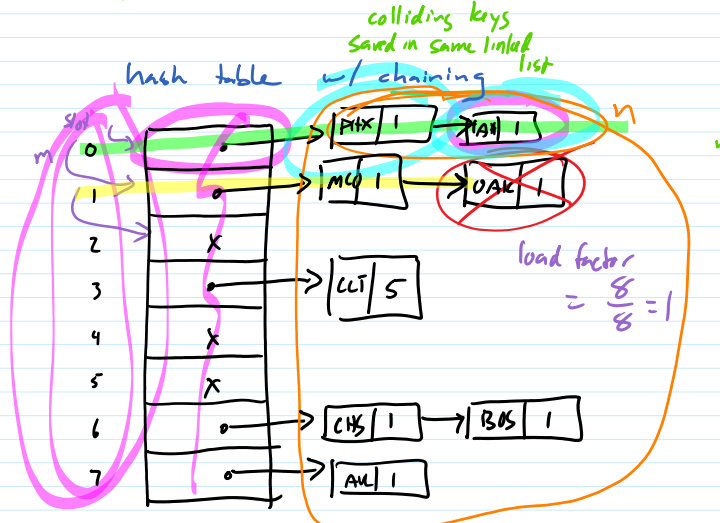
contains-key: compute hash(key) % size  
 if NULL at that location, return NO  
 if something there, do comparison

AVL 65207 % 8 = 7  
 CHS 66702 % 8 = 6  
 IAH 72240 % 8 = 0  
 PHX 79200 % 8 = 0  
 ↳ % 16 = 0

collision w/ BOS  
 ↓  
 resolve collisions on put

```

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

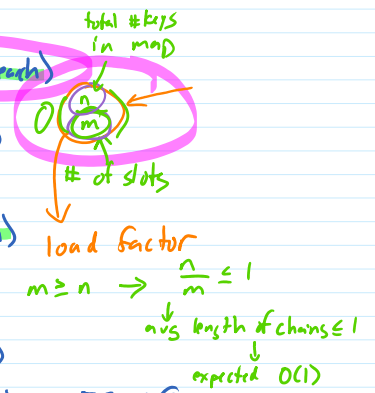


CLT  
 AVL  
 IAH  
 PHX  
 OAK  
 CLE  
 BOS  
 MCO

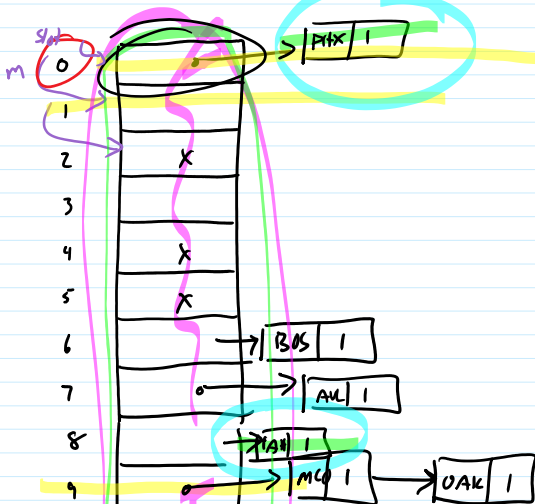
contains key: compute hash(key) % # slots  
 search linked list at that index (sequential search)

get: compute hash(key) % # slots  
 search linked list at that index (sequential search)  
 return value in node

put: compute hash(key) % # slots  
 search linked list at that index (sequential search)



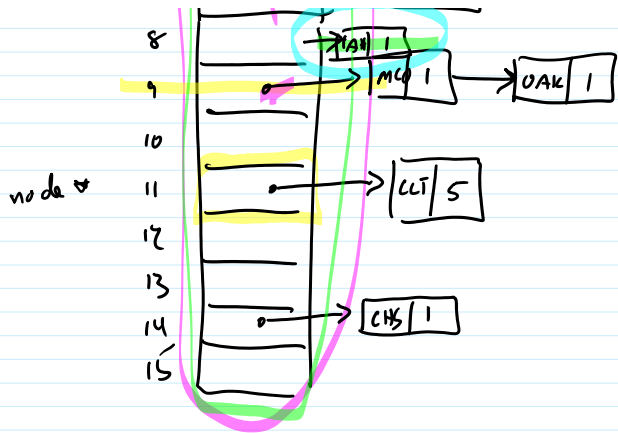
hash fn should distribute keys evenly



contains-key(OAK)

$hash(OAK) = 78009$   
 $78009 \% 16 = 9$

expected running time =  $O(\text{load factor})$   
 if  $m \geq n$   
 $= O(1)$



$= O(1)$

worst case  $O(n)$   
(all keys collide in same slot)

```

20144987 4 4 4 CLT OAK X 17 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
201442094 3 4 4 CHS CLT - 16 1.00 X
20144214 3 4 4 IAH CLT - US 1.00 X
201441020 1 4 4 AVL CLT - 16 1.00 X
201441110 4 4 4 PHX OAK X US 1.00 X
20144794 3 4 4 CLE CLT - 16 1.00 X
201441578 4 5 4 BNA PHL - YX 1.00 X
201442030 4 4 4 DCA BHM X US 1.00 X
34985738957 4 4 4 DCA HVN X US 1.00 X
  
```

$h(\text{CLT}) = 66827 \quad 66827 \% 8 = 3$

$h(\text{OAK}) = 78009 \quad 78009 \% 8 = 1$

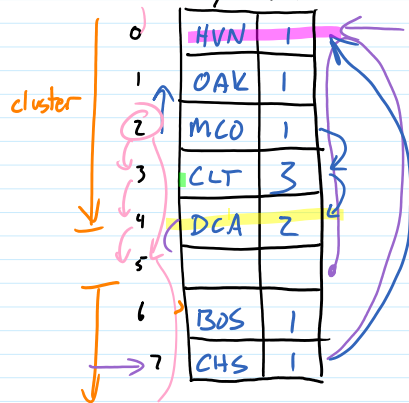
$h(\text{BOS}) = 65958 \quad 6 \quad 6$

$h(\text{MCO}) = 76153 \quad 1$

$h(\text{DCA}) = 67490 \quad \% 8 = 2$

$h(\text{CHS}) = 66702 \quad \% 8 = 6$

hash table w/ open addressing  
linear probing

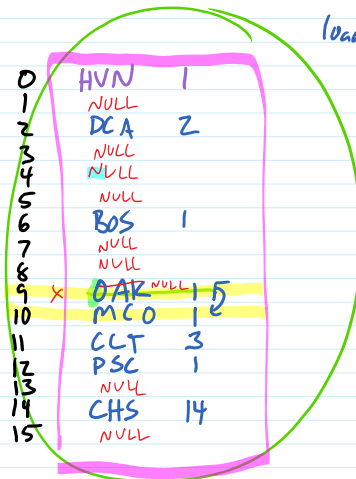


double hash  
2nd hash fun determines  
sequence of slots to search

- contains key  
put  
get
- 1) compute hash
  - 2) compute %
  - 3) sequential search through slots until key found or empty slot or wrap all the way around

never been used  
(not currently occupied and didn't have something taken deleted)

rehash



load factor  $\alpha = \frac{n}{n}$

if  $\alpha \leq \frac{1}{2}$  (half full)

avg cluster  $\approx 2$

expected  $O(1)$  time

$h(\text{PSC}) = 9$

$9 \% 16 = 9$

remove OAK

find MCO