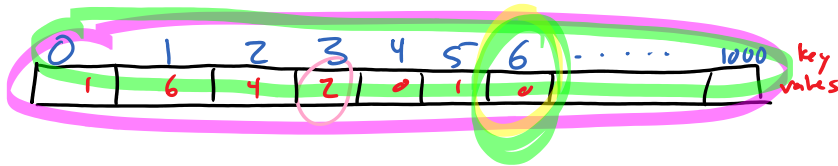


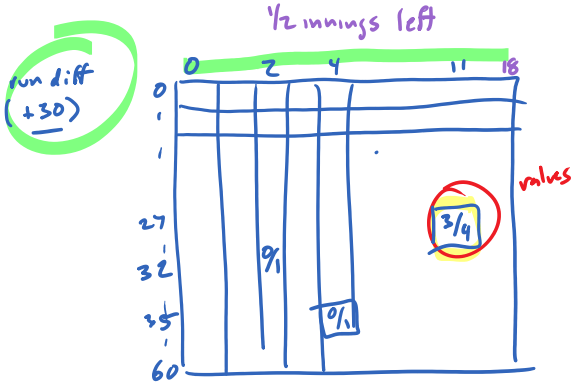
siblings frequency



day of week

key	0	1	2	3	4	5	6	key values
day	Sun	Mon	Tue	Wed	Thu	Fri	Sat	
freq	1	1	0	0	1	3	3	

compute $P(\text{win} \mid \text{lead, half innings left})$



key	lead	innings	w/L
2	8	L	
5	7	L	
-3	3.5	W	
-3	3.5	L	
-3	3.5	W	
-3	3.5	W	

Trek Scott Trek Schwinn P...

airports
 000 →
 HVN →
 ZZZ →

map.get("Trek")
 return 3

Bike Count

key	value
Trek	11
Scott	1
Diamondback	11
Schwinn	1

key	domain name	values	ip address
	hck.zoo.cs.yale.edu		128.36.232.50
	mps.google.com		
	grizzly.zoo.cs.yale.edu		128.36.232.18

Map ADT: associates keys w/ values (dictionary)

key=word value=def

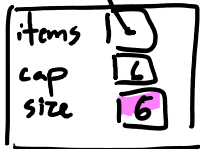
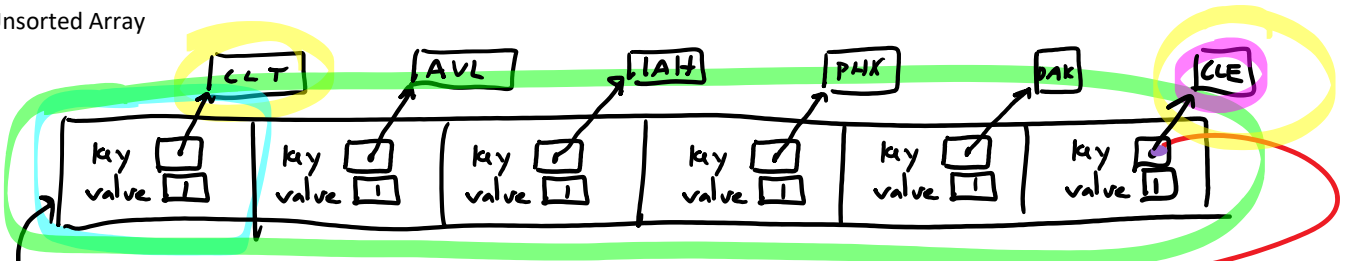
contains-key: determine if key appears in map

get: retrieve value associated with key

put: associate a value with a key, replacing old value if key already associated w/ something

key value
 ACU1294631 Master Bedroom
 ACU1299432 Basement

Map as Unsorted Array



```
struct entry {
    char *key;
    int value;
};
```

```
map-put(m, airport, 1)
```

```
airport → CLT CLT
strcpy(airport, "CLT")
```

contains key

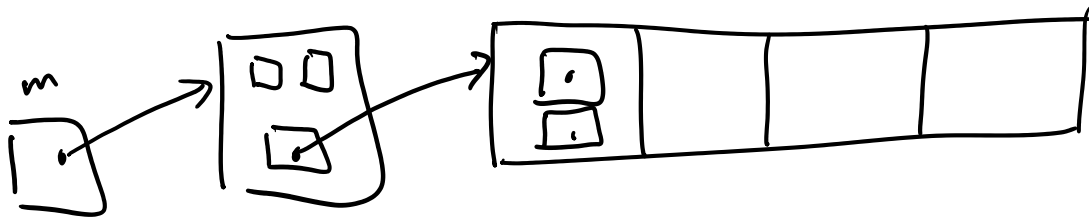
sequential search for key
 $O(n)$ worst case

get sequential search for key
 $O(n)$ worst case

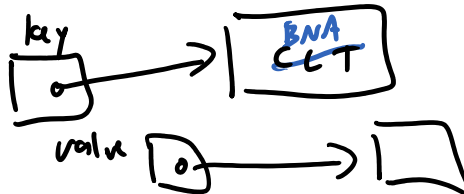
put sequential search
 $O(n)$ worst case

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

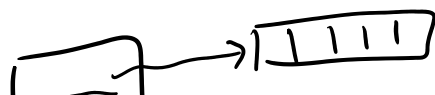
$O(n^2)$ total time
worst case to count n things

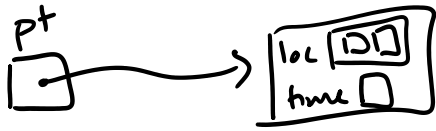
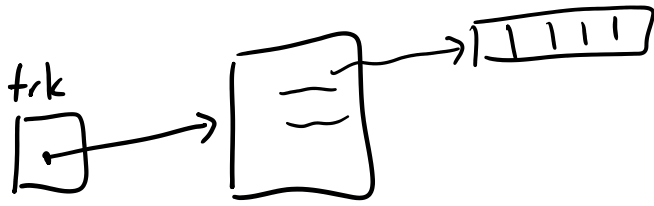


```
char *dest = -
strcpy(dest, "CLT");
map-put(m, dest);
```



trk





`track-add-point(trk, pt)`

Map as Hash Table

```

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 function

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

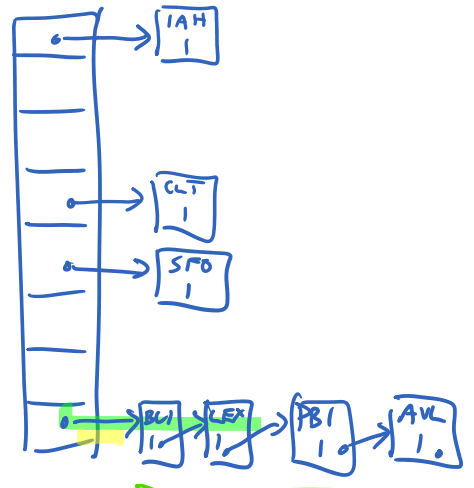
PBI → 78999 % 8 = 7 collision !!!

LEX → 75263

SFO → 82012

hash table

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



smmap-put(m, "PBI", 1)

smmap-contains-key
 :
 worst case $O(n)$
 average case $O(1)$
