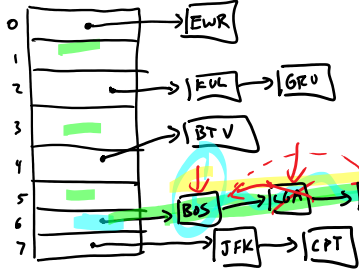
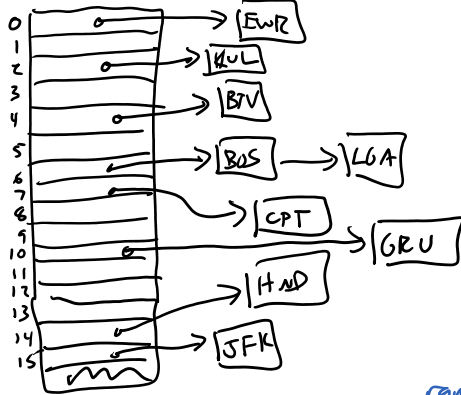


load factor $\alpha = \frac{1}{8}$

- JFK 73359 7
- BOS 65958 6
- EWR 69088 0
- LGA 75302 6
- BTV 66116 4
- HND 71678 6
- KUL 74786 2
- CPT 66951 7
- GRU 70858 2

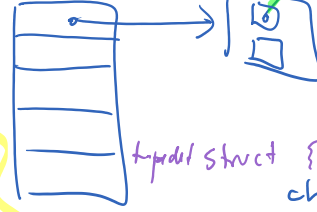


- JFK 73359 15
- BOS 65958 6
- EWR 69088 0
- LGA 75302 6
- BTV 66116 4
- HND 71678 14
- KUL 74786 2
- CPT 66951 7
- GRU 70858 10



random 5 characters

JFK
xg149 JFK
AVL
xg149 AVL



```
typedef struct {
    char * code; ✓
    char * state; ✓
} airport;
```

airport = destroy(...)

```
hash(const char * key)
{
```

```
    const char * temp
    = malloc(strlen(key) + 6);
```

```
    strcpy(temp, key);
```

```
    strcat(temp, key);
```

```
    return normal-hash(temp);
}
```

open addressing w/ linear probing

- JFK 73359 7
- BOS 65958 6
- EWR 69088 0
- LGA 75302 6
- BTV 66116 4
- HND 71678 6
- KUL 74786 2
- CPT 66951 7
- GRU 70858 2

put(JFK, 52)
put(BOS, 47)

0	EWR	57
1	LGA	46
2	HND	67
3	KUL	99
4	BTV	99
5	CPT	104
6	BOS	47
7	JFK	52

bound α by $\frac{1}{2}$

0	EWR	57
1		
2		
3		
4		
5		
6	BOS	47
7	LGA	46
8	CPT	104
9	HND	
	JFK	52

```
smap - ...
smap - remove(m, "LGA");
smap - contains-key(m, "CPT");
```

1000 slots

900 key/value entries

expected ^{probes} time to know key not present

$\frac{1}{1-\alpha}$