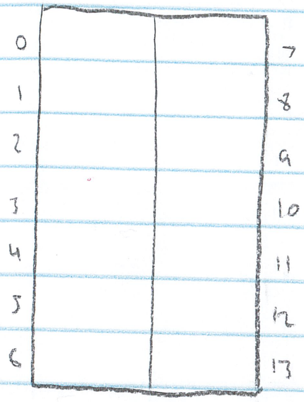
Alg Hashing 11/7 T



- ☆ Unique
- ☆ range
- ☆ Uniform distribution

Good Hash Design Practice

Design to have half of hash table filled

5 1 3 7 4 5 3 0 0 0

x primes > 9 % size

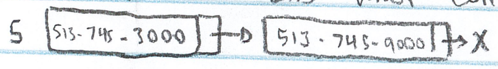
Probably unique

May not be unique

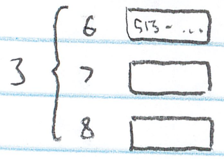
Collision → Calculated key for insert points to occupied space.

Solutions:

Open-hash → Create a LinkedList when collision happens



Bucketing →



key points to bucket of mem. locations

Clustering → Many inputs/results hit same spot

- Usually has snowball affect and becomes worse
- Can improve situation by shifting

(!) shift algorithm must be able to hit all of m

Shif by an $a \in \mathbb{Z}$ s.t. $a \nmid m$ where m: size of table

Alg

11/7/17

"Best" solution →

① Hash to key

② Calculate offset value/Factor

- Still suffers from secondary collision

"Double Hashing"

Hashing Big-O

Insert • O(1)

Search • O(1)

Deleting • (!) Oh no, not simple but still O(1), why?

Tombstone → Leave a Flag "something was here before"

- This solves problem of broken offset paths