Hash maps

$h(k) = (ak+b) \% p$

p - prime

$a \in \{1, \ldots, p-1\}$

$b \in \{0, \ldots, p-1\}$

hash functions
works well ⇒ keys are uniformly distributed across slots.

Other hash functions in common use
Multiplicative hash fn; division hash,
modern: SipHash, ChaCha hash

hashing with charning (open hashing)

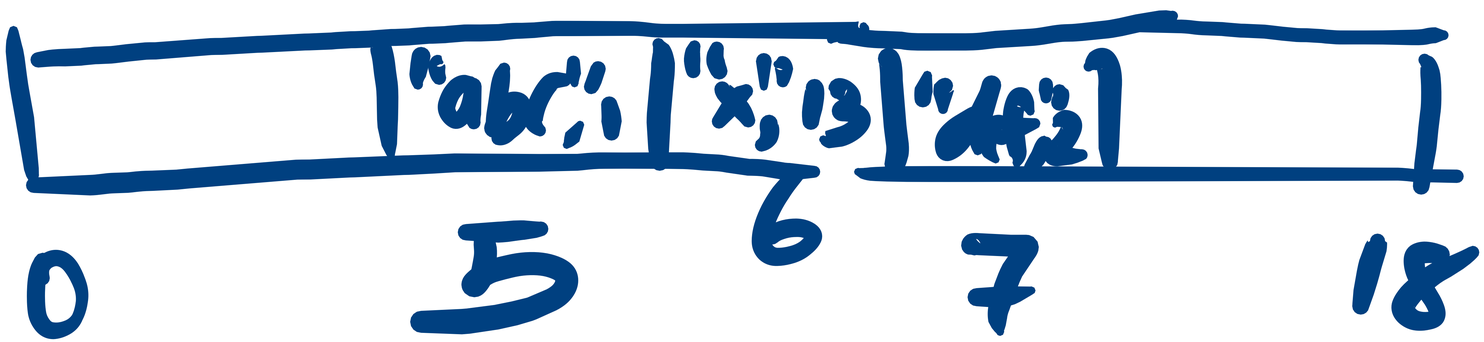
Closed hashing

Say "abc" and "def" has the same hash value.

In closed hashing, we do not use linked lists. We store (key, value) in the array itself..

"abc", "def" → 5

("abc", 1)
("def", 2)



Rehashing

- keep a count of number of elements in the hashmap, say n.
- if h.p * load_factor < n

  Set h.p to 2^a prime number in [n, 2n]

elements in h

rehash(h)

h.p = random prime in [n, 2n]

h.a = random int in (1, h.p-1]

h.b = random int in [0, h.p-1]

slots = new vector of size h.p

for each k,v in h

insert(slots, k, v)

deallocate h.slots.

let n be # elements

let $p = n/2$

Assume hashes distributes elts uniformly.

Let X_i be the length of linked list in slot i.

$\mathbb{E}[X_i] = 2$

$P(X_i \geq 16) \leq 1/8$

$P(X_i \geq 32) \leq 1/16$