#ASLI ENGINEERING

Double Hashing in Hash Tables

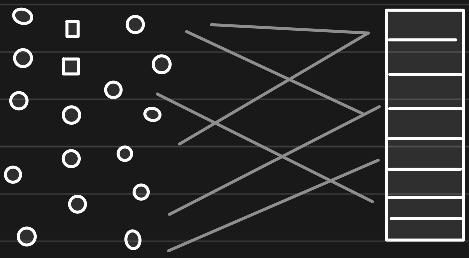
SWIPE

BY

ARPIT BHAYANI

Conflict Resolution using Double Hashing

Conflicts are inevitable!



With Open Addressing, we use a probing function to find the slot where the key should be placed

One such method is Double Hashing

Probing Function

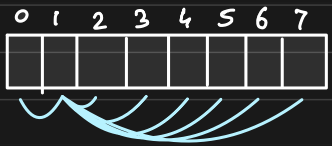
Probing Function is defined as $\overset{\text{key}}{\underset{\downarrow}{}} \; p(k, i) = j$ ← index

key ↘ ↙ attempt

we use the probing function to find the first available slot

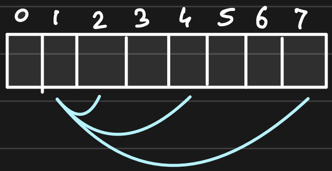
the same function is used during lookups

Linear and Quadratic Probing



$p(k,i) = h(k) + i$

linear



$p(k,i) = h(k) + c_1 i + c_2 i^2$

Quadratic

ARPIT BHAYANI

Double Hashing

Double Hashing techniques 2 hash functions to find the slot

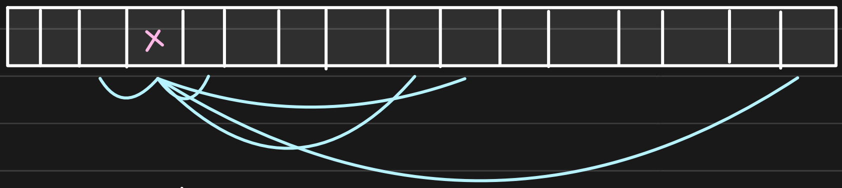
First hash function gives primary slot

and upon collision, it uses second hash function as offset
times attempt

until an empty slot is reached

$$p(k,i) = \left(h_1(k) + i * h_2(k) \right) \bmod m$$

Given that we are using another hash function to offset, we are minimizing repeated collisions and effects of clustering



Follows no specific pattern
and gives near-uniform yet random offset from the index (primary slot)

Sequence: $h_1(k)$, $h_1(k) + 1 \cdot h_2(k)$, $h_1(k) + 2\, h_2(k)$,

$h_1(k) + 3\, h_2(k)$, $h_1(k) + 4\, h_2(k)$, ...

Choosing the second hash function

1. It should never return 0
2. It should cycle through entire table
   (order does not matter)
3. Fast to compute and ∼ to a random number generator

$p(k,i) = h_1(k) + i * \underline{h_2(k)}$

Advantages of Double Hashing

1. uniform spread upon collision
2. follows no specific offset pattern
   ↳ purely depends on the key
3. least prone to clustering problem
   ↳ offset from primary slot is uniformly distributed