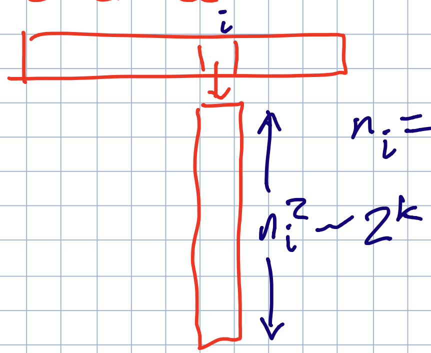
algorithms.wtf

Hashing

Chained:



$n_i = |h^{-1}(i)|$

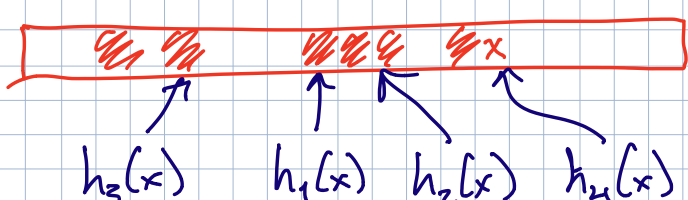
$O(1)$ worst-case time

$O(n)$ expected space

assuming universal hashing

$\Pr[h(x)=h(y)] \le \frac{O(1)}{m}$

Open addressing



Probe sequence for x

$h_1(x)\ h_2(x), h_3(x) \dots$

Lookup(x):

for $i \leftarrow 1$ to m
    if $T[h_i(x)] = x$
        TRUE
    else if $T[h_i(x)] = \emptyset$
        FALSE

~~Ideal:~~
~~For all x:~~
~~$(h_1(x), h_2(x), \ldots)$ is a random perm of $(0, 1, \ldots, m-1)$~~

not possible

$$\Rightarrow E[\text{time for a search}] = O\left(\frac{m}{m-n}\right)$$

n items
m slots

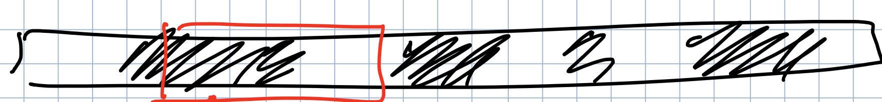
$m-n$ empty

$$\Pr[\text{empty}] = \frac{m-n}{m} = 1 - \frac{n}{m}$$

$$E[\#\text{probes}] = \frac{m}{m-n}$$

Linear probing: $h_i(x) = (h(x) + i) \bmod m$

unfortunate/unintuitive clumping

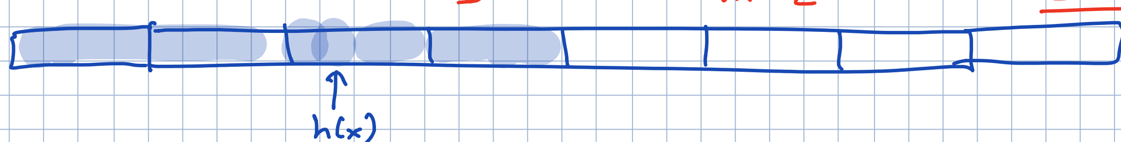


cache line!

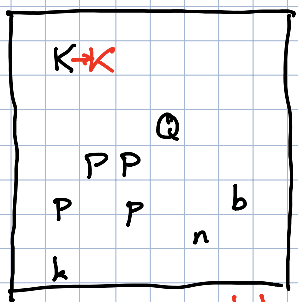
Binary probing:
$m = 2^\ell$

$$\boxed{h_i(x) = h(x) \oplus i}$$

$E[\text{time}] = O(1)$ if $\frac{n}{m} < \frac{1}{2}$ and h is 5-uniform



$\Pr[h(a)=i \quad h(b)=j \quad h(c)=k \quad h(d)=\ell \quad h(e)=\ell'] \leq \frac{1}{m^5}$



ZOBRIST HASHING

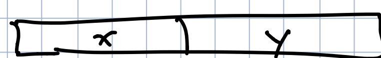
$h = T[K@b7] \oplus T[k@b1] \oplus \cdots$

$T[\ldots]$ filled with random #s

$h' = h \oplus T[K@b7] \oplus T[K@c7]$

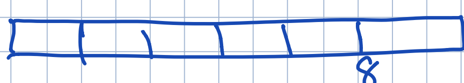
Carter Wegman '76 Tabulation hashing

$\mathcal{U} = [2^w]$ $\quad (x,y) \in \mathcal{U} \quad x \in [2^{w/2}] \quad y \in [2^{w/2}]$



$\left.\begin{array}{l} A[0 \ldots 2^{w/2}-1] \\ B[0 \ldots 2^{w/2}-1] \end{array}\right\}$ ideal random integers

$$h(x,y) = (\underbrace{A[x]}_{\text{random}} \oplus \underbrace{B[y]}_{\text{random}}) \bmod m$$



$$h(x_1 \ldots x_8) = \bigoplus_{i=1}^{8} A_i[x_i]$$

$\boxed{\text{3-uniform not 4-uniform}}$

$$h(x,y) \oplus h(x',y) \oplus h(x,y') \oplus h(x',y') = 0$$

But almost ideal random

[PatrascuThorup05]

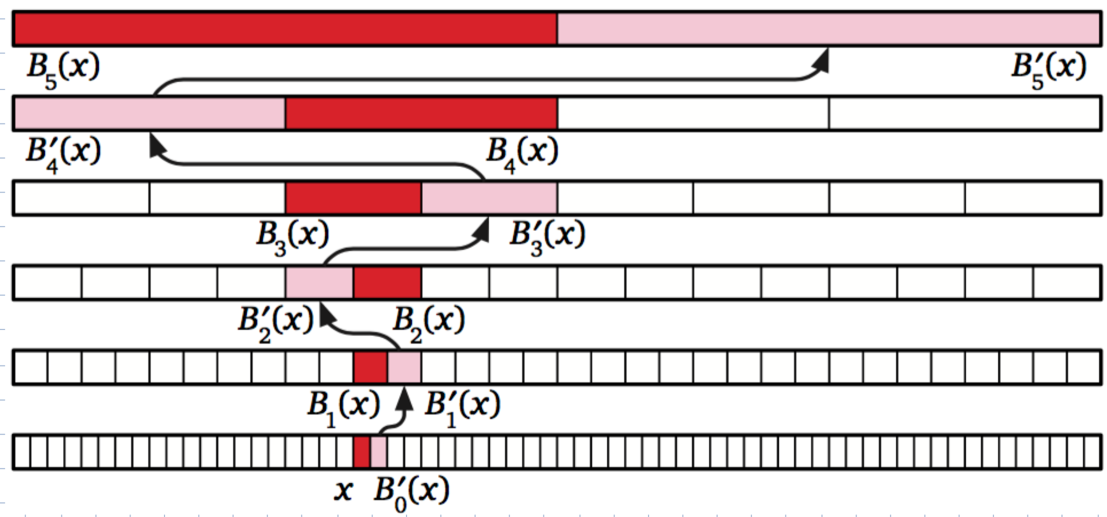
Good enough for linear probing

Modified Tab Hashing [Thorup Zhang 12]

$$h(x,y) = A[x] \oplus B[y] \oplus C[x+y]$$

5-uniform!

k-uniform
c chunks $\Rightarrow$ $\sim c \cdot k$ table lookups



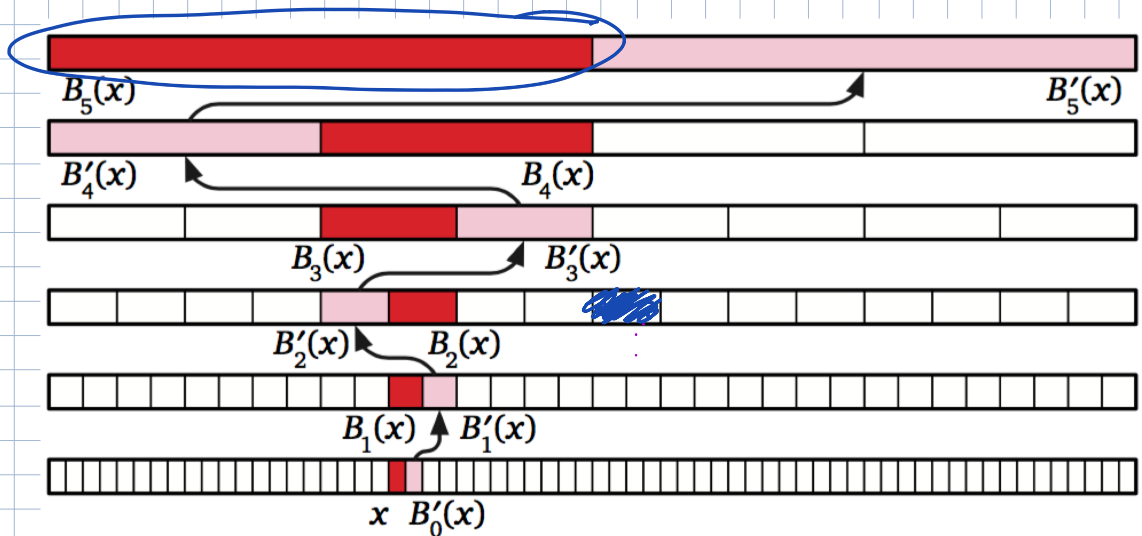
ancestors
$B_k(x)$ = block of size 2^k containing $h(x)$

uncles
$B'_k(x)$ = sibling of $B_k(x) = B_{k+1}(x) \setminus B_k(x)$

```
LooseBinaryProbe(x):
  if H[h(x)] = x
      return True
  if H[h(x)] is empty
      return False
  first ← Dunno

  for k ← 0 to ℓ − 1
      for each index j ∈ B'_k(x) in arbitrary order
          if H[j] = x
              first ← True
          if H[j] is empty
              first ← False
      if first ≠ Dunno
          return first
```

$$E[\text{time}] \le O(1) + \sum_{k=1}^{\ell-1} O(2^k) \cdot \Pr[B'_k \text{ is } \underline{\text{full}}]$$



B_k is full $\iff$ B_k is popular $\leftarrow$ 2^k items hashed here

or one of B_k's uncles is ~~full~~ popular

Need $\Pr[B'_k(x)$ is popular$]$

$\boxed{\text{Assume } n = \frac{m}{2}}$

$E[\#\text{items hash into } B_k] = 2^{k-1}$

$\Pr[\#\text{items} \geq 2 \cdot E[\#\text{items}]]$

5-indep $\Rightarrow$ 4^{th} moment inequality

$\Pr[\text{popular}] \leq O(4^{-k})$

$$E[\text{time}] = \sum_k O(2^k) \cdot \Pr[B_k \text{ is popular}]$$

$$= \sum_k O(2^k) \cdot O(4^{-k}) = \sum_k O(2^{-k})$$

$$= O(1)$$

Cuckoo hashing

x is stored at $T[h(x)]$ or $T[h'(x)]$

use tabulation hashing