LRU with larger cache

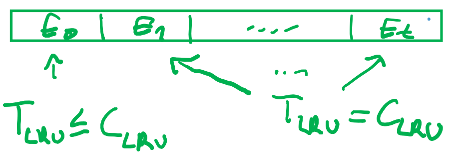
What if we give LRU an advantage of larger cache size $C_{LRU} > C_{OPT}$?

Thm: (Sleator, Tarjan) For every $C_{LRU} > C_{OPT} \geq 1$ and every access sequence,

$$T_{LRU} \leq \underbrace{\frac{C_{LRU}}{C_{LRU} - C_{OPT}}}_{\text{competitive ratio} > 1} \cdot T_{OPT} + C_{OPT}. \quad (*)$$

advantage of initial cache state for OPT

Pf: Split the access sequence into "epochs" $E_0, \dots, E_t$ s.t. LRU misses are:



What are OPT misses in each epoch?

1) Epoch E_i for $i > 0$ $\implies$ in E_i: $T_{LRU} \leq \frac{C_{LRU}}{C_{LRU} - C_{OPT}} \cdot T_{OPT}$

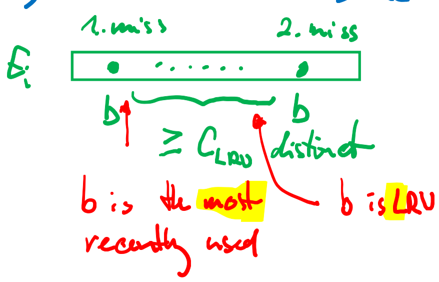
   a) LRU misses on distinct blocks in E_i

   $\implies$ E_i contains $\geq C_{LRU}$ distinct blocks

   $\implies$ OPT misses at least $C_{LRU} - C_{OPT}$ times in E_i

   at most C_{OPT} in cache at the start of E_i

   b) LRU misses on same block b ≥ 2 times in E_i



   $\implies$ $\geq C_{LRU}$ accesses to distinct blocks between misses (not necessarily misses)

   $\implies$ OPT misses at least $C_{LRU} - C_{OPT}$ times in E_i

2) Epoch E_0: $T_{OPT} \geq T_{LRU} - C_{OPT}$ (LRU misses on distinct blocks, up to C_{OPT} can be saved)

   $\implies$ in E_0: $T_{LRU} \leq \underbrace{\frac{C_{LRU}}{C_{LRU} - C_{OPT}}}_{>1} \cdot T_{OPT} + C_{OPT}$

By summing over all epochs, we obtain $(*)$. $\square$

Cor: For $C_{LRU} = 2\,C_{OPT}$, LRU is $(2+\varepsilon)$-competitive on long enough sequences.

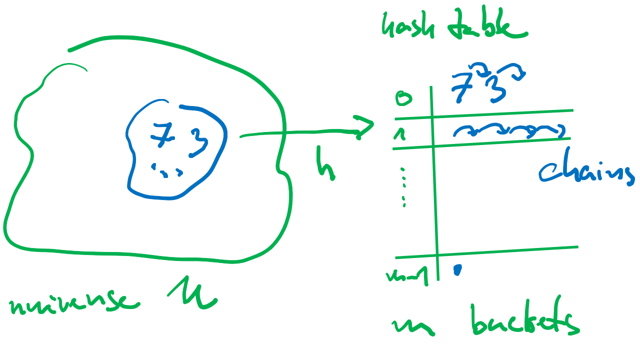
to mitigate the term C_{OPT}

Note: I/O complexity of our algorithms is (at worst) linear in cache size M

$\implies$ having LRU with cache size M instead of OPT with $M/2$ preserves asymptotics.

Hashing (with chaining)

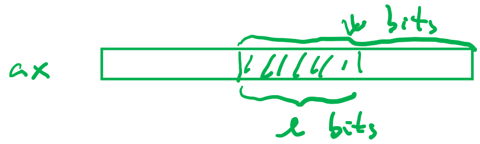
goal: set representation with $O(1)$ average time operations and $O(n)$ space.
(expected time) ↑ FIND, INSERT, DELETE



hash function $h: \mathcal{U} \to [m] = \{0, \dots, m-1\}$
universe size U
table size (# buckets) m
set size (# items) n
$\Rightarrow$ n/m expected chain length if h is totally random
"density"

ex: • linear congruence
$h(x) = ax \bmod m$, where a, m coprime

• multiply - shift
$h(x) = \lfloor (ax \bmod 2^w) / 2^{w-\ell} \rfloor$ where $m = 2^\ell$, a odd, $w > \ell$



• scalar product
$\underbrace{h(x_1, \dots, x_d)}_{x \in \mathbb{Z}_p^d} = \sum_i a_i x_i \bmod m = a \cdot x \bmod m$ where $a \in \mathbb{Z}_p^d$
in $\mathbb{Z}_p$
for hashing of strings

• polynomial
$h(x_1, \dots, x_d) = \sum_i a^i x_i \bmod m$ where $a \in \mathbb{Z}_p$

☹: For any of these hash functions, if x is uniformly distributed $\Rightarrow$ $h(x)$ uniformly distributed $\Rightarrow$ n/m expected chain length (for random input)

Q: What if we have an adversarial input sequence?

A: Choose h randomly from some family $\mathcal{H}$ of functions.
with uniform distribution

ex: $\mathcal{H}$ = all functions $h: \mathcal{U} \to [m]$ (totally random function)

$\Pr_{h \in \mathcal{H}}[h(x) = h(y)] = 1/m$ for any $x \neq y$

choice of h, not x, y! "collision"

But h needs $O(U \log m)$ bits!
no need for hashing if we have U bits!

What is a "good" hashing family?

- parameterized – $O(1)$ space for hash function
- $O(1)$ time for computation ($O(d)$ for strings of length d)
- "behaves as a totally random function" (if $h \in \mathcal{H}$ picked uniformly in random)

Def: A family $\mathcal{H}$ of functions $h: \mathcal{U} \to [m]$ is c-universal for some $c > 0$ if

$$\Pr_{h \in \mathcal{H}} [h(x) = h(y)] \leq \frac{c}{m} \quad \text{for any } x \neq y \text{ from } \mathcal{U}.$$

over choice of h collision at most c-times worse than totally random f.

Note: $\mathcal{H}$ is universal if $\mathcal{H}$ is c-universal for some c.

Thm: Let $\mathcal{H}$ be a c-universal family of $h: \mathcal{U} \to [m]$, $\overbrace{x_1, \dots, x_n}^{\text{already hashed}}, y \in \mathcal{U}$ distinct. (new item)

Then $$\mathbb{E}_h [\# i : h(x_i) = h(y)] \leq \frac{cn}{m}.$$

expected chain length

Pf: $A = \sum_i A_i$ indicator random variable $A_i = \begin{cases} 1 & h(x_i) = h(y) \\ 0 & \text{else} \end{cases}$

$$\mathbb{E}[A_i] = 0 \cdot \Pr[A_i = 0] + 1 \cdot \Pr[A_i = 1] = \Pr[A_i = 1] = \Pr[h(x_i) = h(y)] \leq \frac{c}{m}$$ c-universal

$$\mathbb{E}[A] = \mathbb{E}[\sum_i A_i] = \sum_i \mathbb{E}(A_i) \leq \frac{cn}{m}. \quad \square$$

linearity of expectation

Complexity of hashing with chaining

Assume h picked uniformly from c-universal $\mathcal{H}$, $x_1 \dots x_n$ already hashed. (distinct)

- FIND(y) ... $O(\frac{cn}{m})$ expected time if unsuccessful (y distinct)
- INSERT(y) ... — " — (assuming we do not know if y is there)
- FIND(y) ... $O(\frac{cn'}{m})$ expected time if successful ($y = x_i$ for some i)
  where n' = # hashed items when x_i inserted (added to the end of chain)
- DELETE(y) ... — " —

$\Rightarrow$ $O(1)$ expected time if $m = \Omega(n)$.

Hashing with flexible m (when n unknown at start)

if $\alpha = n/m > 1$, then $m' = 2m$ + rehash all items (sketch)

if $\alpha < 1/4$, then $m' = m/2$ + —‖—

$\Rightarrow$ similar to flexible arrays, keeps $\alpha \in [1/4, 1]$ ← or other thresholds

$\Rightarrow$ $O(1)$ expected amortized time (rehashing amortizes to $O(1)$)

↪ amortization of expected time

Constructions of universal families

- scalar product:

$h_t(x) = x \cdot t$ for $x, t \in \mathbb{Z}_m^d$ (d-dim. vector space over the field $\mathbb{Z}_m$ if m prime)

Thm: $\mathcal{H} = \{ h_t \mid t \in \mathbb{Z}_m^d \}$ is 1-universal for any m prime, $d \geq 1$.

Pf: $x, y \in \mathbb{Z}_m^d$ distinct, assume w.l.o.g. $x_d \neq y_d$

$$\Pr[h_t(x) = h_t(y)] = \Pr[x \cdot t = y \cdot t] \overset{\text{bilinearity of scalar product}}{=} \Pr[(x-y) \cdot t] = \Pr\Big[\sum_{i=1}^{d} (x_i - y_i) t_i = 0\Big]$$

$$= \Pr\Big[\underbrace{(x_d - y_d)}_{\neq 0} t_d = -\sum_{i=1}^{d-1} (x_i - y_i) t_i\Big] = 1/m$$

After choice of $t_1, \dots, t_{d-1}$ there is exactly one $t_d \in \mathbb{Z}_m$ s.t. (linear equation over field $\mathbb{Z}_m$)

Since $t_1, \dots, t_{d-1}$ picked uniformly at random $\Rightarrow \sum_{i=1}^{d-1} (x_i - y_i) t_i$ is also uniformly random (except when $x_i = y_i$ for every $i < d$)

$\Rightarrow \Pr[t_d$ is the solution of $] = 1/m$ $(\Pr[t_d = 0] = 1/m)$ $\square$

Cor: (d=1) $\{ h_a \mid a \in \mathbb{Z}_m \}$ is 1-universal for any m prime.

$$h_a(x) = ax \pmod{m}$$

What if we want to hash into $[m]$ with general m? (not prime) linear congruence

Let $p \geq m$ be a prime. For $a, b \in \mathbb{Z}_p$ we define

$$h_{ab}(x) = ((ax+b) \bmod p) \bmod m$$

Note: For flexible m we can find p prime $m < p \leq 2m$. (Bertrand's postulate)

Thm: The family $\mathcal{H}' = \{ h_{ab} \mid a, b \in \mathbb{Z}_p \}$ is 2-universal for any prime $p \geq m$.