

Module 3: Hashing

Lecture 10: Linear hashing

Prev topic

Next topic

Prev page

Next page

The Lecture Contains:

Linear hashing

- Principle
- Searching

Module 3: Hashing

Lecture 10: Linear hashing

[Prev topic](#)[Next topic](#)[Prev page](#)[Next page](#)

Linear hashing

- Number of buckets grow by at most 1
 - Linear growth
- Both primary and overflow buckets
 - Overflow buckets are chained
- Family \mathcal{g} of hash functions $\{h_0, \dots, h_i, \dots\}$
 - $h_i(k) = h(k) \bmod (2^i n)$
 - n is initial number of buckets
 - h_{i+1} doubles the range of h_i

Module 3: Hashing

Lecture 10: Linear hashing

Prev topic

Next topic

Prev page

Next page

Linear hashing

- Number of buckets grow by at most 1
 - Linear growth
- Both primary and overflow buckets
 - Overflow buckets are chained
- Family \mathcal{G} of hash functions $\{h_0, \dots, h_i, \dots\}$
 - $h_i(k) = h(k) \bmod (2^i n)$
 - n is initial number of buckets
 - h_{i+1} doubles the range of h_i
- Load factor decides when to split
- Split pointer s decides which bucket to split
 - s is independent of overflowing bucket
 - At level i , s is between 0 and $2^i - 1$
 - s is incremented and if at end, is reset to 0
- Records in splitting bucket are rehashed using h_{i+1}
 - Equal chance of being in old and new buckets

Module 3: Hashing

Lecture 10: Linear hashing

[Prev topic](#)[Next topic](#)[Prev page](#)[Next page](#)

Principle

- Full buckets are not necessarily split
- Buckets split are not necessarily full

Prev topic

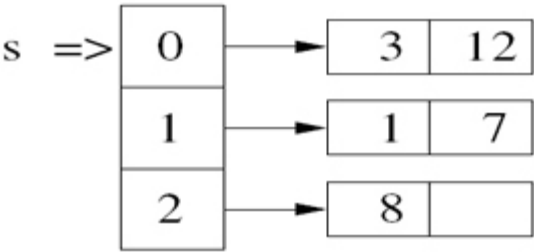
Next topic

Next page

Prev page

Principle

- Full buckets are not necessarily split
- Buckets split are not necessarily full
- Principle: Every bucket will be split sooner or later and so all overflows will be reclaimed and rehashed



Prev topic

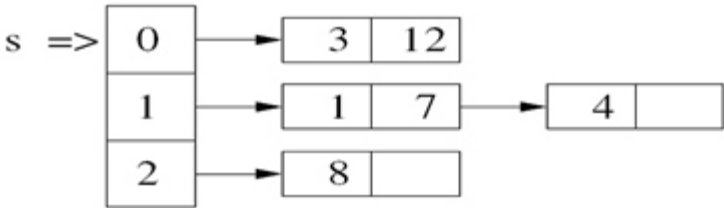
Next topic

Next page

Prev page

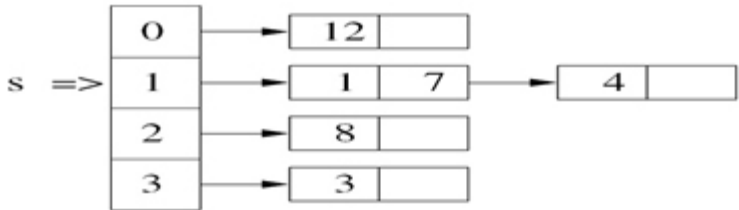
Principle

- Full buckets are not necessarily split
- Buckets split are not necessarily full
- Principle: Every bucket will be split sooner or later and so all overflows will be reclaimed and rehashed



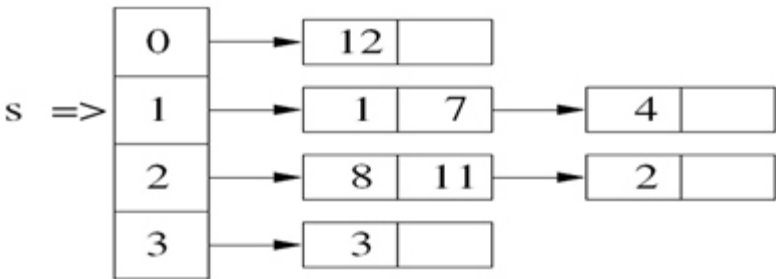
Principle

- Full buckets are not necessarily split
- Buckets split are not necessarily full
- Principle: Every bucket will be split sooner or later and so all overflows will be reclaimed and rehashed



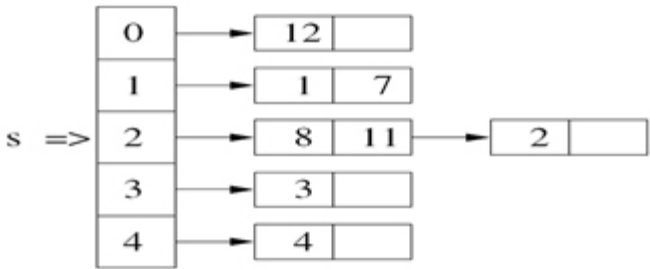
Principle

- Full buckets are not necessarily split
- Buckets split are not necessarily full
- Principle: Every bucket will be split sooner or later and so all overflows will be reclaimed and rehashed



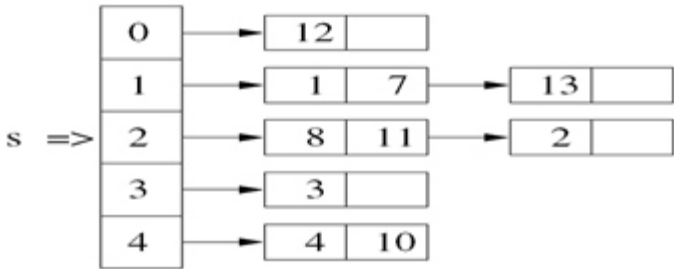
Principle

- Full buckets are not necessarily split
- Buckets split are not necessarily full
- Principle: Every bucket will be split sooner or later and so all overflows will be reclaimed and rehashed



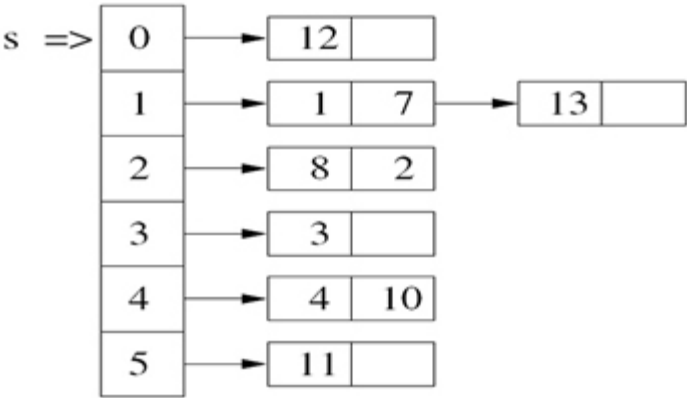
Principle

- Full buckets are not necessarily split
- Buckets split are not necessarily full
- Principle: Every bucket will be split sooner or later and so all overflows will be reclaimed and rehashed



Principle

- Full buckets are not necessarily split
- Buckets split are not necessarily full
- Principle: Every bucket will be split sooner or later and so all overflows will be reclaimed and rehashed



Module 3: Hashing

Lecture 10: Linear hashing

[Prev topic](#)[Next topic](#)[Prev page](#)[Next page](#)

Searching

- Searching key r
- Find bucket $b = h_l(r)$, l being final level
- If $b > s$, then bucket b has not been split and r must be here
- Otherwise r may be in b or $b + 2^l$
- Apply $h_{l+1}(r)$ to find out