# Algorithms

## Hashing II

Jiří Vyskočil, Marko Genyk-Berezovskyj

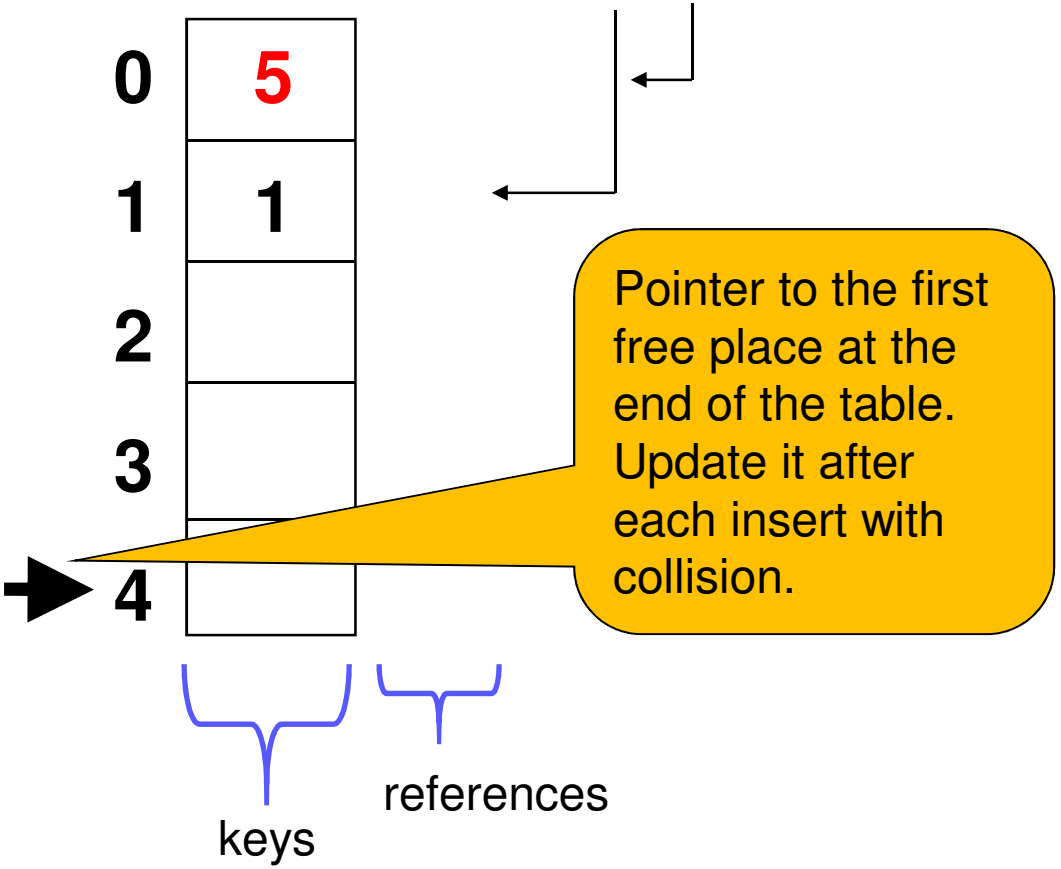
2010

# Coalesced hashing

- Approximate no. of elements is known.
- Store colliding elements elsewhere, typically the collision is produced by non-synonym in the table, ie. by other key which itself collided sometime earlier.
- Equip each table position by a single reference to colliding key if such exists.
- => Colliding elements are stored in the same table. eferences create chains which are subject to so called coalescence.
  - LISCH (late insert standard coalesced hashing)
  - EISCH (early insert standard coalesced hashing)
  - *using additional cellar space*
    - LICH (late insert coalesced hashing)
    - EICH (early insert coalesced hashing)
    - VICH (variable insert coalesced hashing)

# Late insert standard coalesced hashing - LISCH

- $h(k) = k \bmod 5$
- sequence: 1, 5, 21, 10, 15

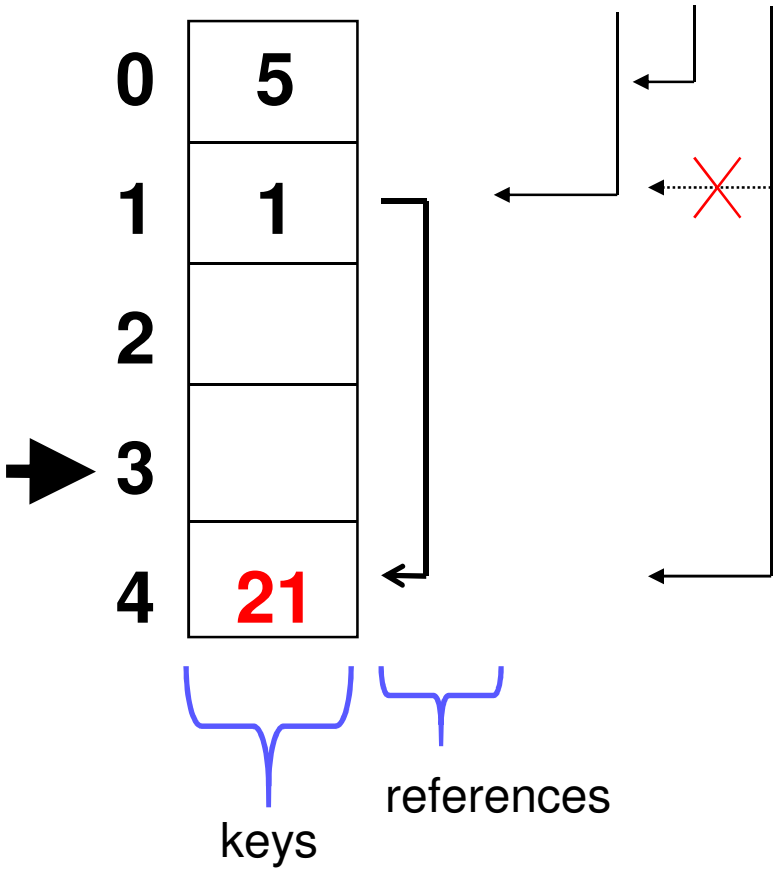



Method:

1. $i = h(k)$;
2. Search the chain starting at position i.
3. If $k$ is not found store it at the table end pointer and also append it to the end of the collision chain.

# Late insert standard coalesced hashing - LISCH

- $h(k) = k \bmod 5$
- sequence: 1, 5, 21, 10, 15

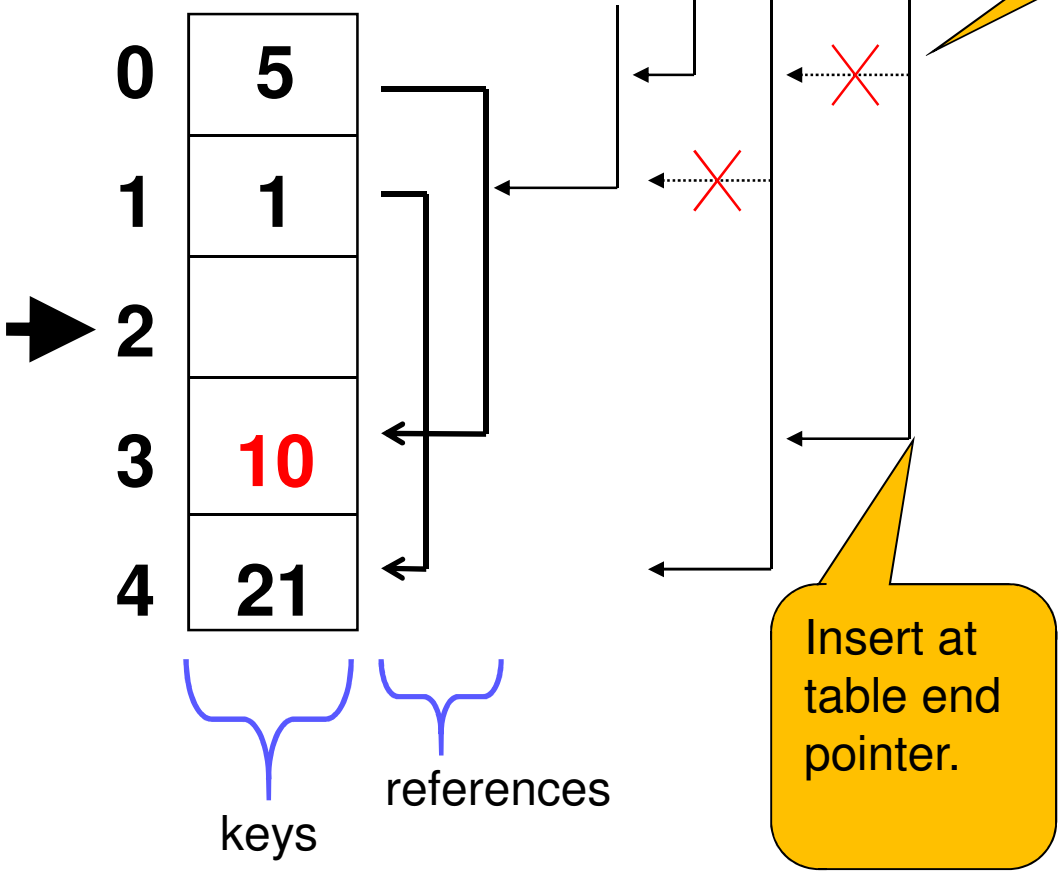



Method:

1. $i = h(k)$;
2. Search the chain starting at position i.
3. If $k$ is not found store it at the table end pointer and also append it to the end of the collision chain.

# Late insert standard coalesced hashing - LISCH

- $h(k) = k \bmod 5$
- sequence: 1, 5, 21, 10, 15




Traverse the chain.

Insert at table end pointer.

Method:

1. $i = h(k)$;
2. Search the chain starting at position i.
3. If $k$ is not found store it at the table end pointer and also append it to the end of the collision chain.

# Late insert standard coalesced hashing - LISCH

- $h(k) = k \bmod 5$
- sequence: 1, 5, 21, 10, 15

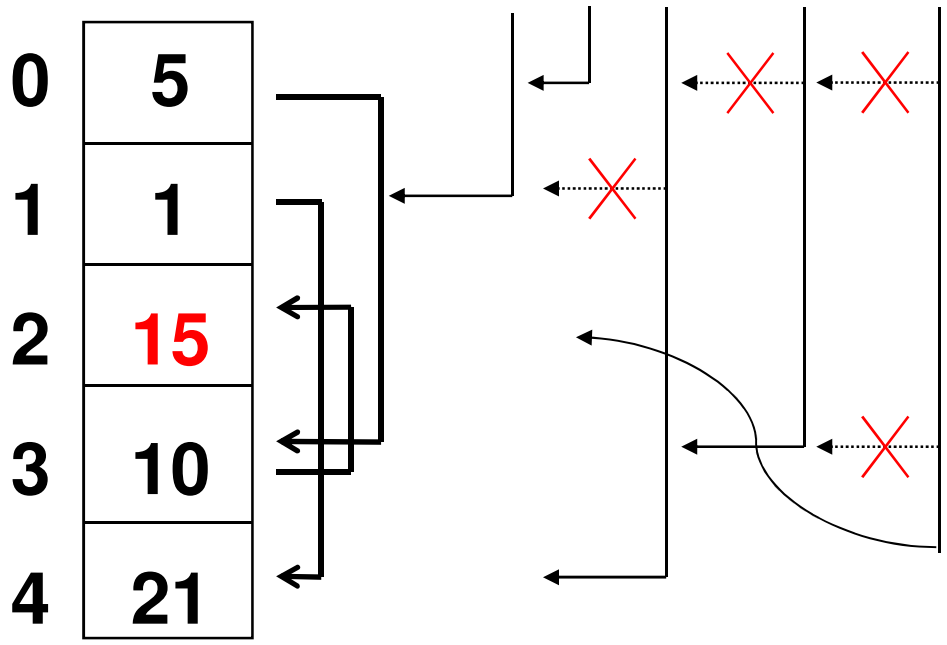


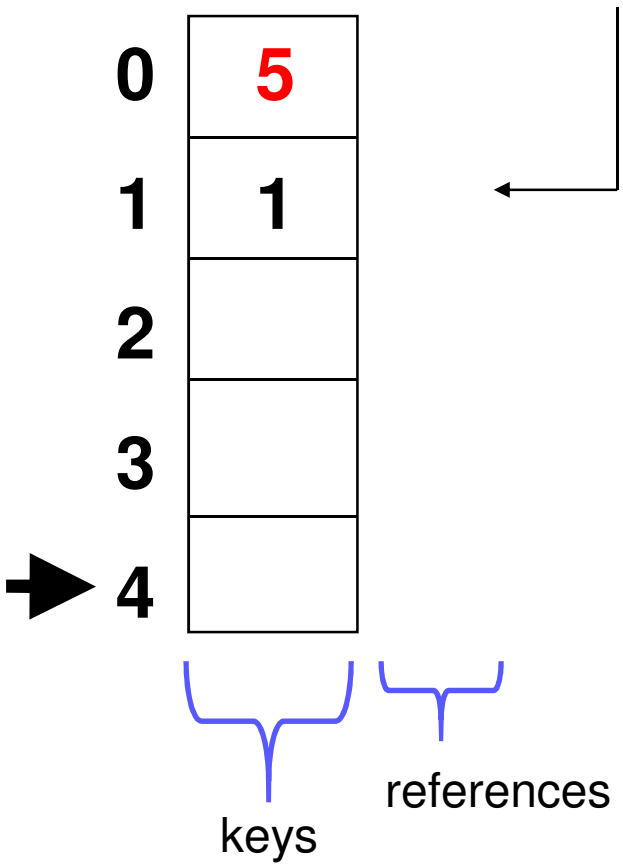

Table is full.

Postup:

1. $i = h(k)$;
2. Prohledej řetězec začínající na místě $i$ a pokud nenajdeš $k$, přidej ho do tabulky na první volné místo od konce tabulky a připoj ho do řetězce na poslední místo.

# Early insert standard coalesced hashing - EISCH

- $h(k) = k \bmod 5$
- sequence: 1, 5, 21, 10, 15

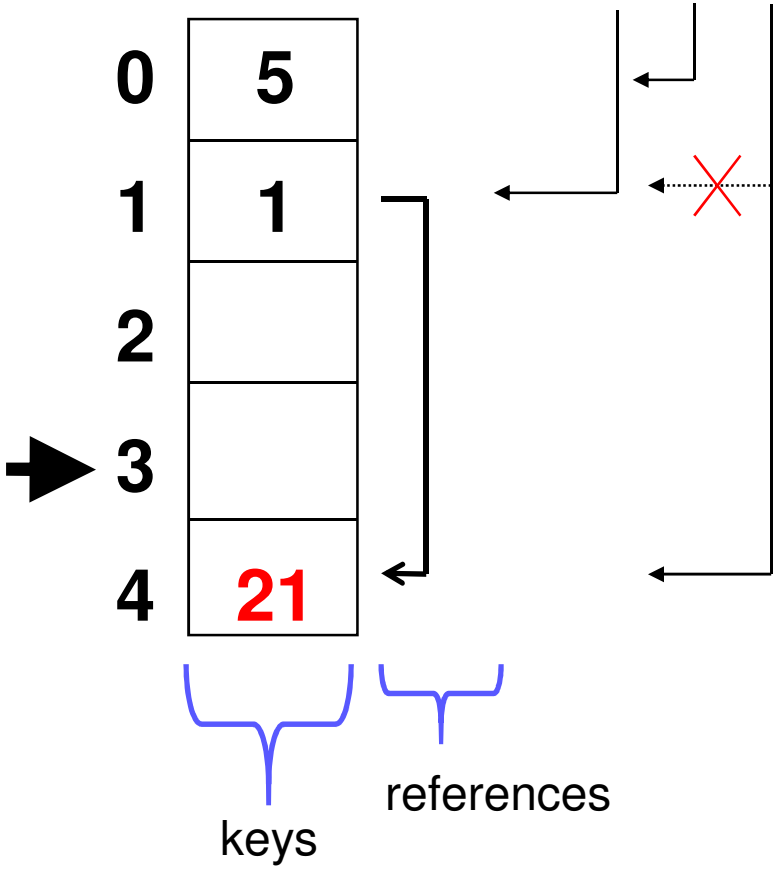



Method:

1. $i = h(k)$;
2. Search the chain starting at position i.
3. If $k$ is not found store it at the table end pointer and also append it just behind the first element of the collision chain.

# Early insert standard coalesced hashing - EISCH

- $h(k) = k \bmod 5$
- sequence: 1, 5, 21, 10, 15

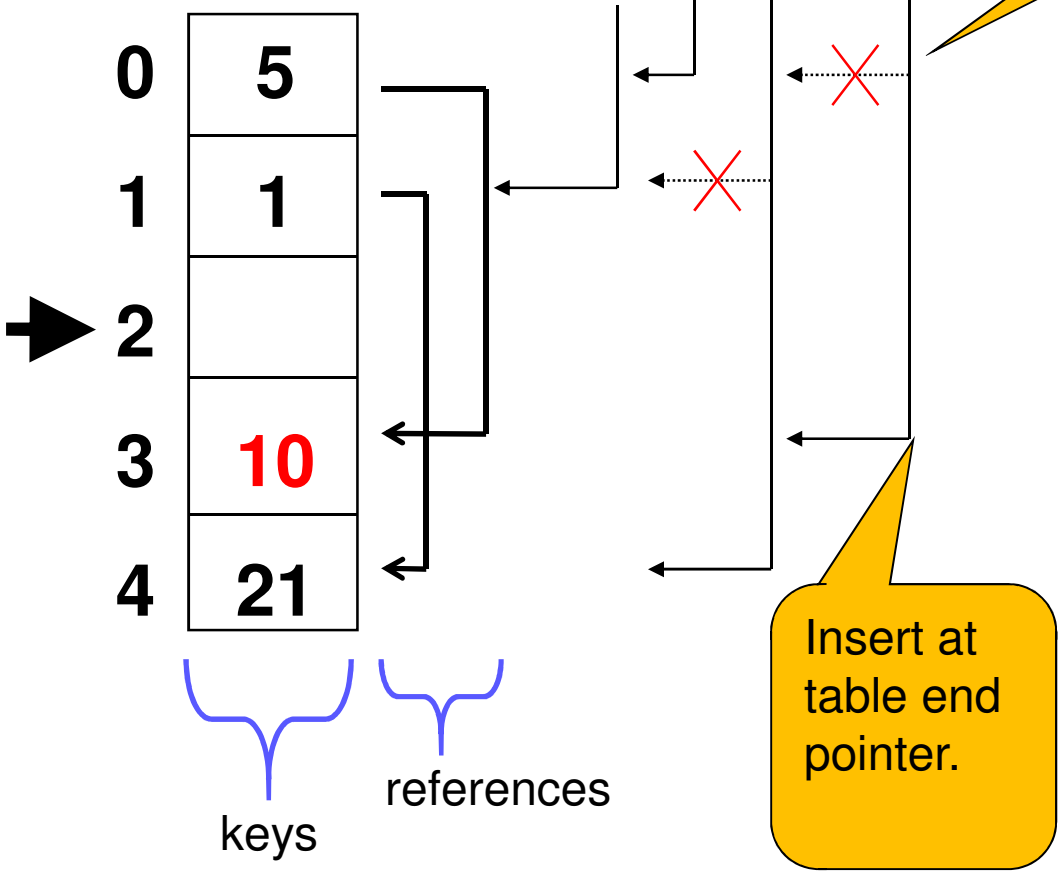



Method:

1. $i = h(k)$;
2. Search the chain starting at position i.
3. If $k$ is not found store it at the table end pointer and also append it just behind the first element of the collision chain.

# Early insert standard coalesced hashing - EISCH

- $h(k) = k \bmod 5$
- sequence: 1, 5, 21, 10, 15




Traverse the chain.

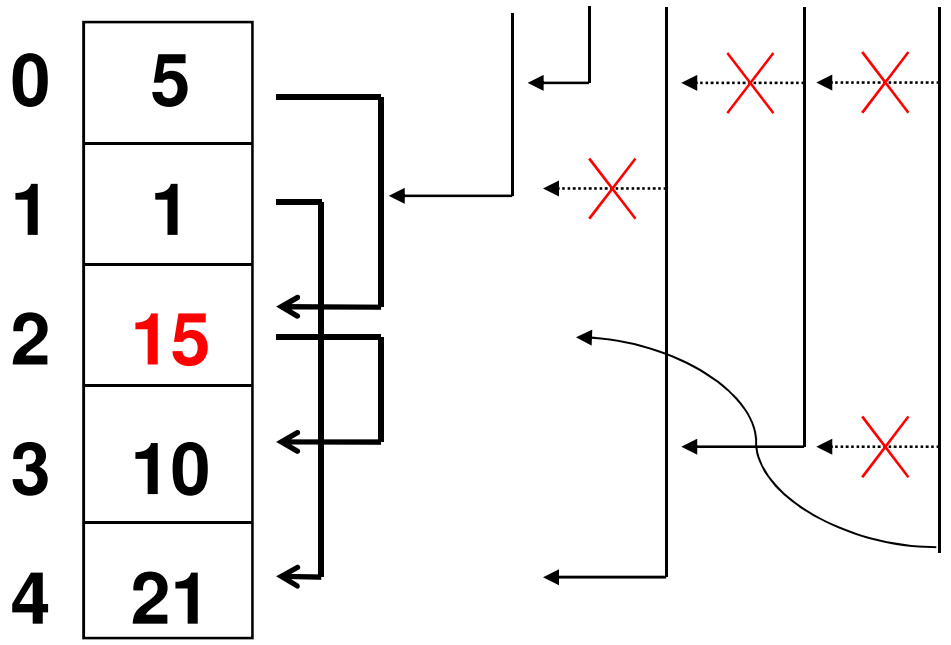
Insert at table end pointer.

Method:

1. $i = h(k)$;
2. Search the chain starting at position i.
3. If $k$ is not found store it at the table end pointer and also append it just behind the first element of the collision chain.

# Early insert standard coalesced hashing - EISCH

- $h(k) = k \bmod 5$
- sequence: 1, 5, 21, 10, 15

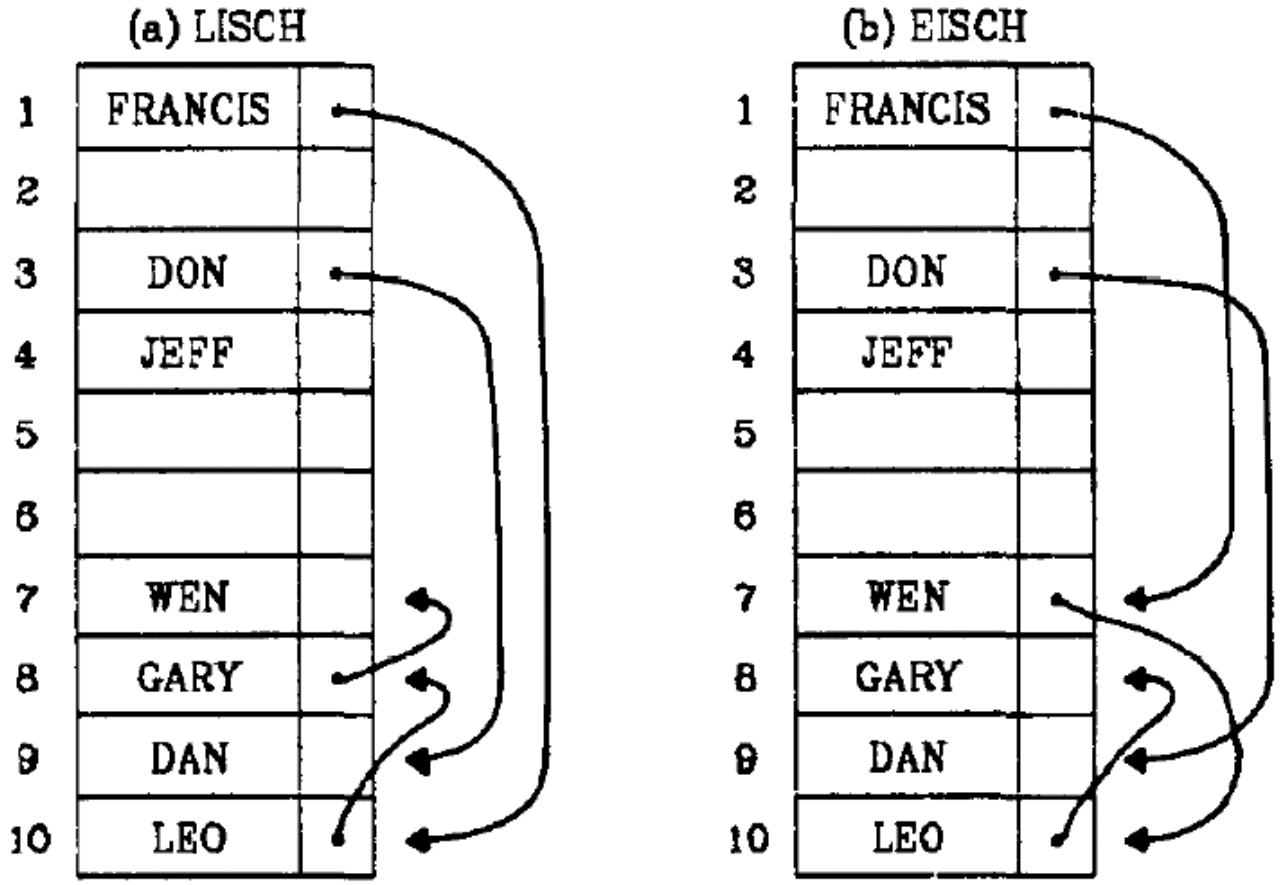



Method:

1. $i = h(k)$;
2. Search the chain starting at position i.
3. If $k$ is not found store it at the table end pointer and also append it just behind the first element of the collision chain.

➡ Table is full.

# Standard coalesced hashing – LISCH, EISCH

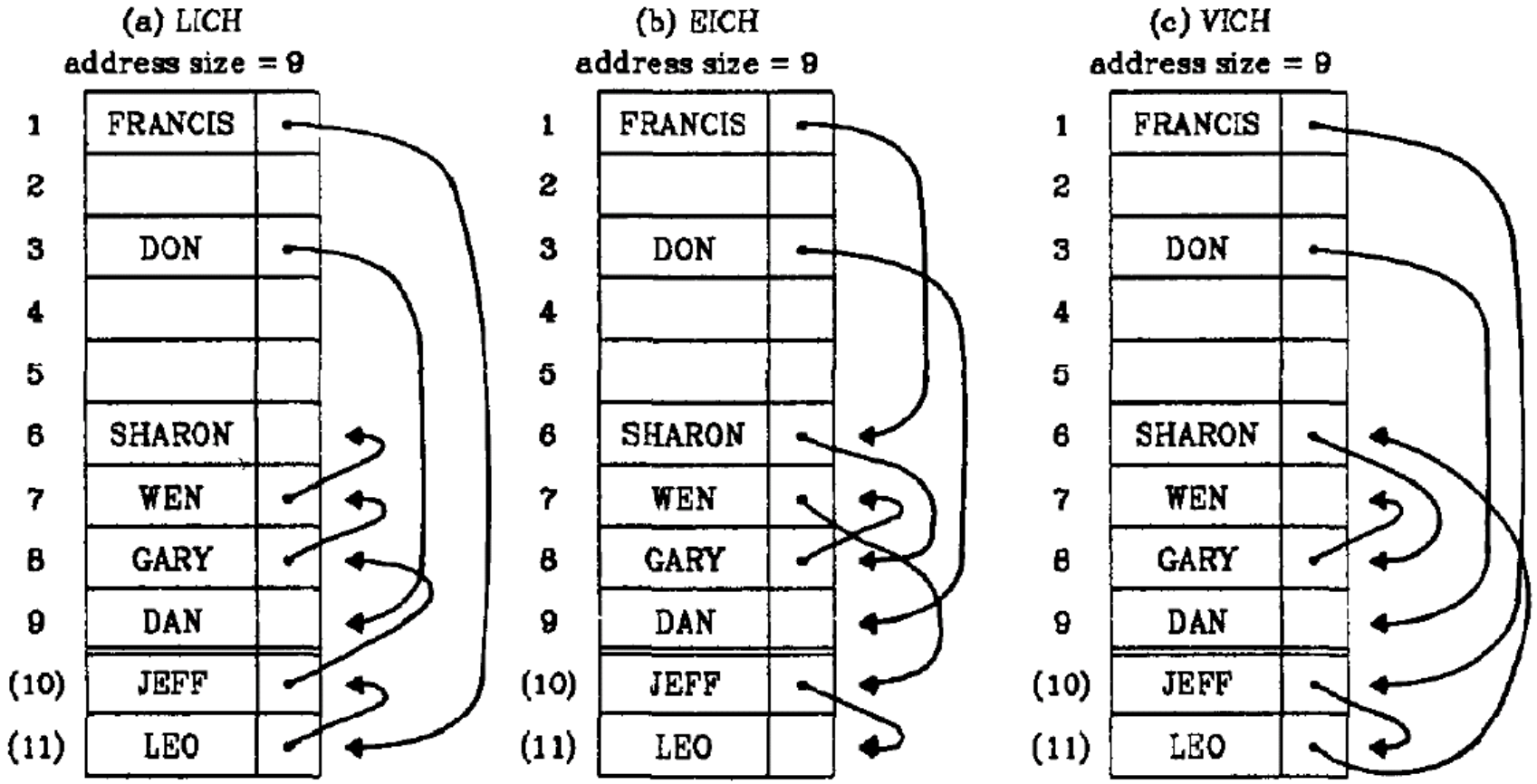



| Keys: | | FRANCIS | DON | LEO | JEFF | DAN | GARY | WEN |
|---|---|---|---|---|---|---|---|---|
| Hash Addresses: | (a)(b) | 1 | 3 | 1 | 4 | 3 | 10 | 1 |

# Coalesced hashing with added cellar space

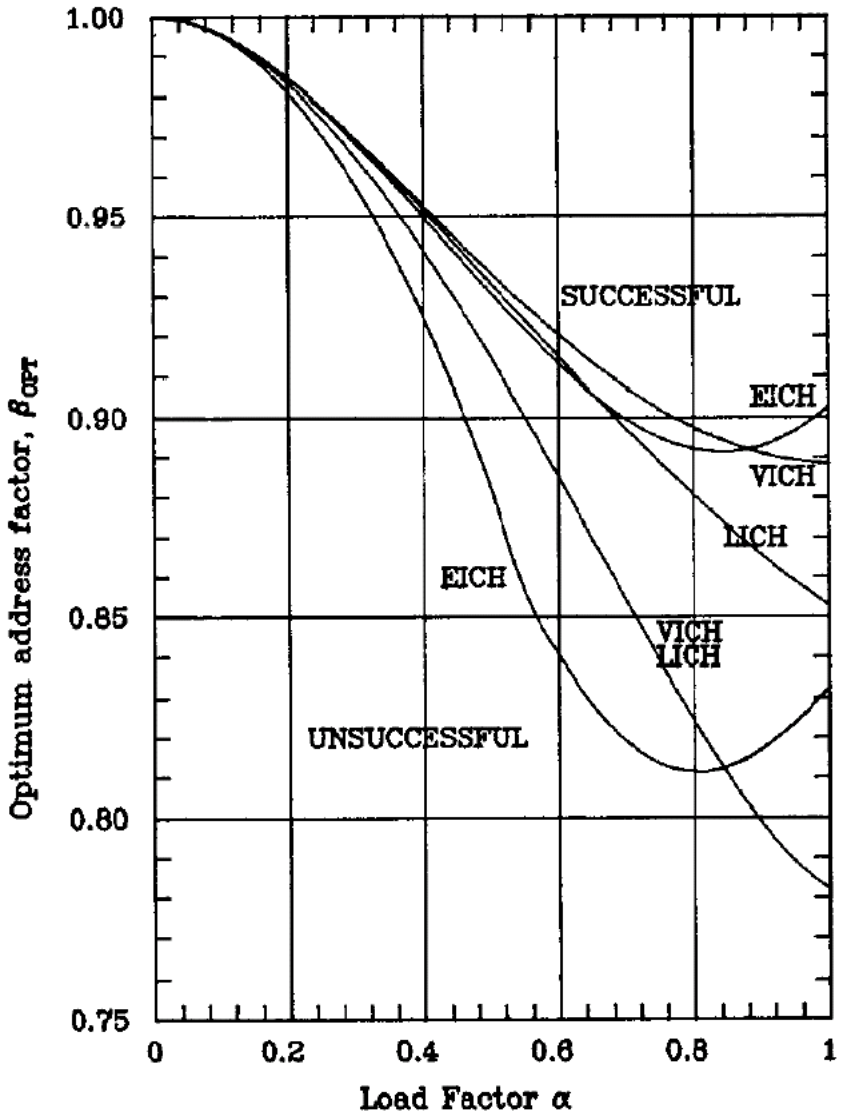
- To reduce the coalescing the table is expanded by non-addressable space - cellar.
- The cellar is at the end of the table and has the same structur as the table.
- Algorithms LICH a EICH are analogous variants of algorithms LISCH a EISCH expanded by the cellar.
- Algorithm VICH (variable insert coalesced hashing) adds the colliding key to the end of collision chain if the chain ends in the cellar. If the chain ends in outside the cellar the key is inserted into the chain at the point where the chain leaves the cellar.

# Coalesced hashing with added cellar space




| Keys: | FRANCIS | DON | LEO | JEFF | DAN | GARY | WEN | SHARON |
|---|---|---|---|---|---|---|---|---|
| Hash Addresses: | 1 | 3 | 1 | 1 | 3 | 1 | 8 | 1 |

# Coalesced hashing with added cellar space



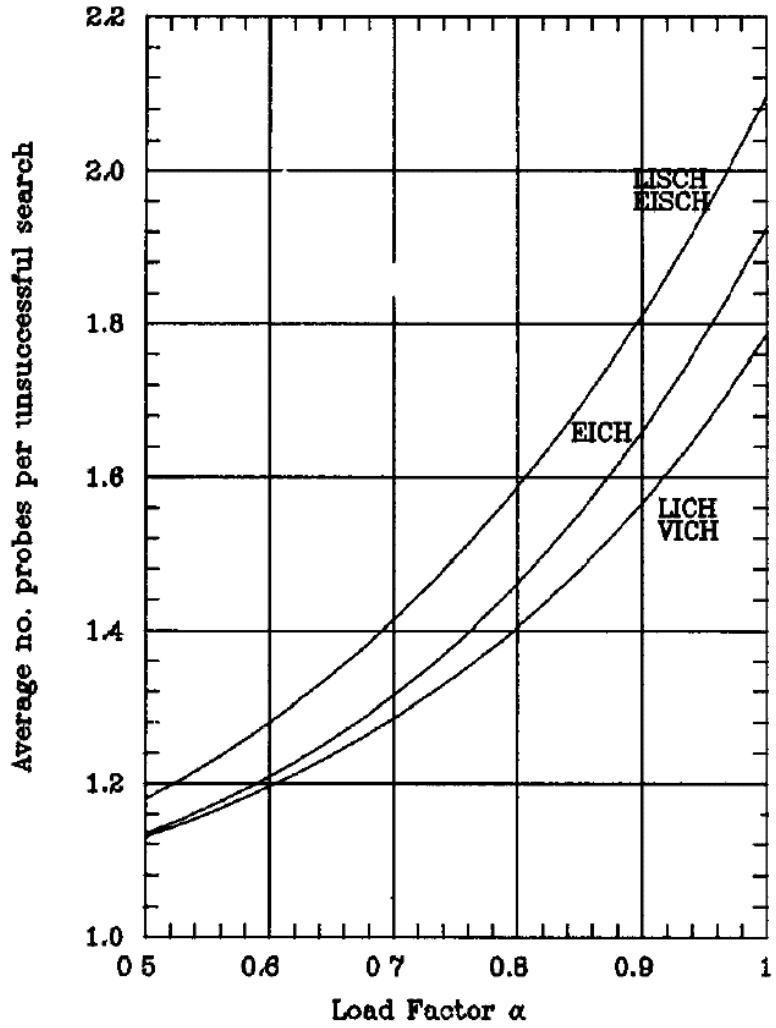

α – load factor
α = N/M'

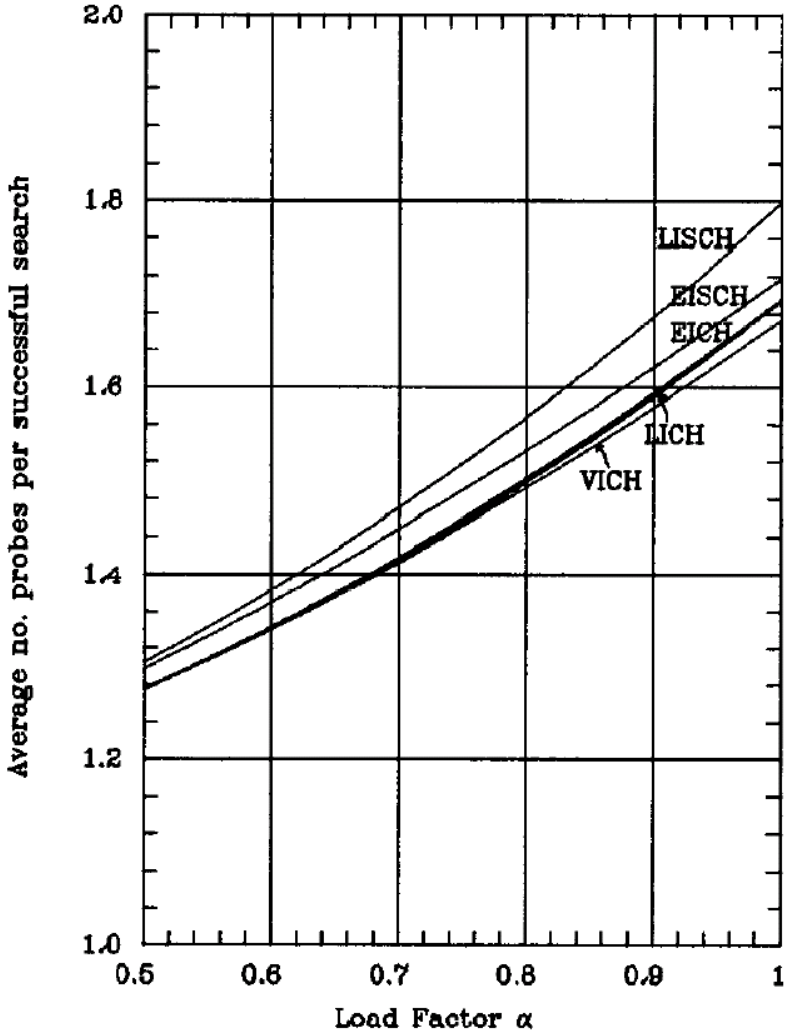
β – address factor
β = M/M'

K = M' – M = cellar size

N – inserted keys
M' – table size
M – addressable table size

# Comparison of coalesced hashing




- If cellar is used VICH performs the best. Recommended β is 0,86.
- If cellar is not used EISCH performs the best.