Cấu trúc dữ liệu và giải thuật

BẢNG BĂM

Giảng viên:
Văn Chí Nam – Nguyễn Thị Hồng Nhung – Đặng Nguyễn Đức Tiến

Nội dung trình bày

- Khái quát về băm
- Hàm băm
- Sự đụng độ
- Các phương pháp giải quyết đụng độ
- Kết luận

© FIT-HCMUS

3

Tìm kiếm theo bảng băm

Hash Table

Khái quát về hash

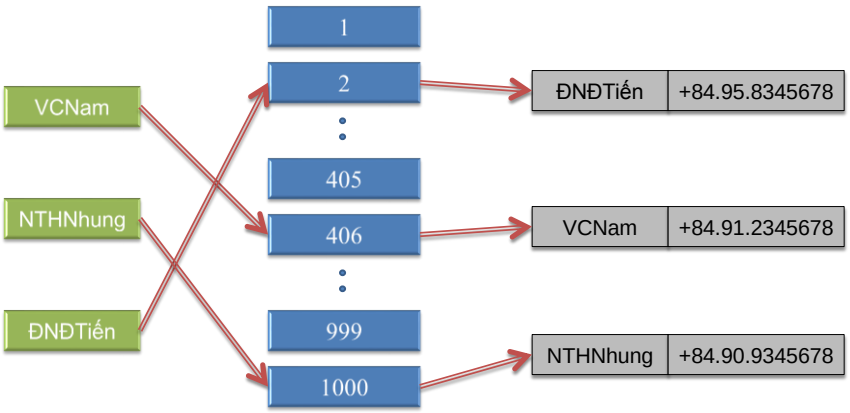
4

- ***Vấn đề:*** Cho trước 1 tập S gồm các phần tử được đặc trưng bởi giá trị khóa. Trên giá trị các khóa này có quan hệ thứ tự. Tổ chức S như thế nào để tìm kiếm 1 phần tử có khóa k cho trước có độ phức tạp ít nhất trong giới hạn bộ nhớ cho phép?
- ***Ý tưởng:*** Biến đổi khóa k thành một số (bằng hàm hash) và sử dụng số này như là địa chỉ để tìm kiếm trên bảng dữ liệu.

© FIT-HCMUS

Ví dụ về một bảng băm

5



Độ phức tạp

6

- Chi phí tìm kiếm trung bình: O(1)
- Chi phí tìm kiếm trong trường hợp xấu nhất: O(n) (rất ít gặp).

© FIT-HCMUS

Hàm băm

7

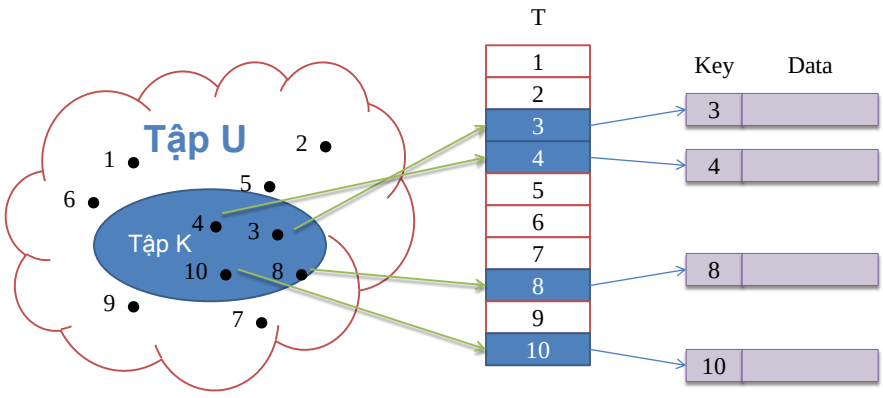
- ***Định nghĩa:*** Hàm băm (hash function) là hàm biến đổi khóa k của phần tử thành địa chỉ trong *bảng băm*.
- ***Tổng quát về phép biến đổi khóa:*** Là 1 ánh xạ thích hợp từ tập các khóa U vào tập các địa chỉ A.

  H: U → A

  k → a = h(k)
- Tập các giá trị khóa (U) có thể lớn hơn rất nhiều so với số khóa thực tế (K) rất nhiều.

Hàm băm

8



Hàm băm

9

- Chọn số (Digit-selection):
  - Chọn một vài chữ số trong khóa và ghép lại tạo thành giá trị băm.
  - Ví dụ:
    - h(00136482**5**) = 35
  - Ưu điểm: Đơn giản, tính toán nhanh
  - Khuyết điểm: Không thể hiện tính chất của khóa, không phân bố đều

Hàm băm

10

- Gấp số (folding)
  - Cộng các chữ số của khóa
  - Nhóm các chữ số thành số và cộng lại
  - Ví dụ:
    - h(001364825) = 0 + 0 + 1 + 3 + 6 + 4 + 8 + 2 + 5 = 29
    - h(**001**364**825**) = 001 + 364 + 825 = 1190

© FIT-HCMUS

Hàm băm

11

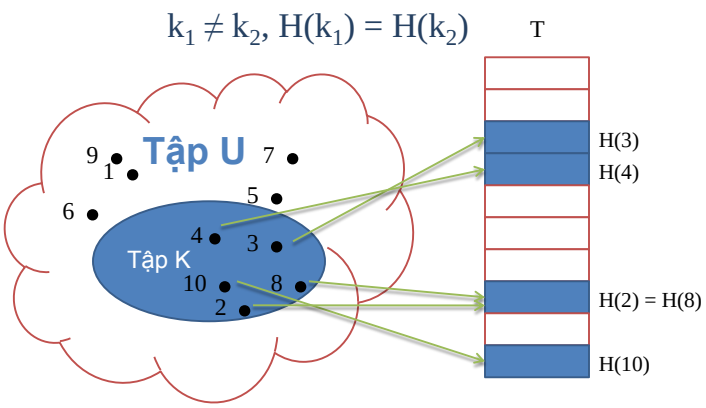
- Lấy dư (modulo arithmetic)
  - Sử dụng phép tính lấy dư
  - h (Key) = Key **mod** tableSize
  - Ví dụ:
    - h(Key) = Key mod 101
    - h(001364825) = 12

Sự đụng độ (collision)

12

- $\exists k_1, k_2 \in K$:

$k_1 \neq k_2, H(k_1) = H(k_2)$



Những yêu cầu đối với hàm băm

13

Các phương pháp xử lý đụng độ

14

- Phương pháp nối kết (separate chaining)
- Phương pháp địa chỉ mở (Open-addressing)

© FIT-HCMUS

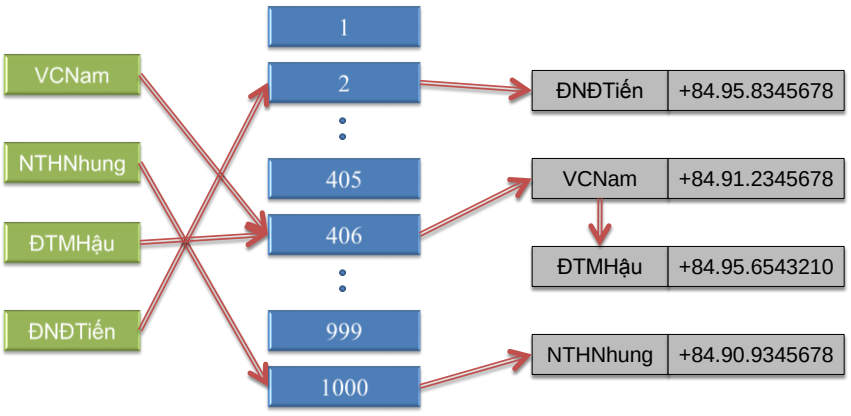
Phương pháp nối kết

15

- Ứng với mỗi địa chỉ của bảng, ta có một danh sách liên kết chứa các phần tử có khóa khác nhau mà có cùng địa chỉ đó.
- Ta sẽ có danh sách (bảng băm) gồm M phần tử chứa địa chỉ đầu của các danh sách liên kết.

Phương pháp nối kết

16



Phương pháp địa chỉ mở

17

- Tên gọi khác:
  - Phương pháp dò
  - Phương pháp thử
- Ý tưởng:
  - Khi đụng độ xảy ra, ta sẽ thử tìm đến vị trị kế tiếp nào đó trong bảng cho đến khi tìm thấy vị trí nào còn trống.

Phương pháp địa chỉ mở

18

- Phương pháp dò tuyến tính (Linear probing)
- Phương pháp dò bậc 2 (Quadratic probing)
- Phương pháp băm kép (Double hashing)

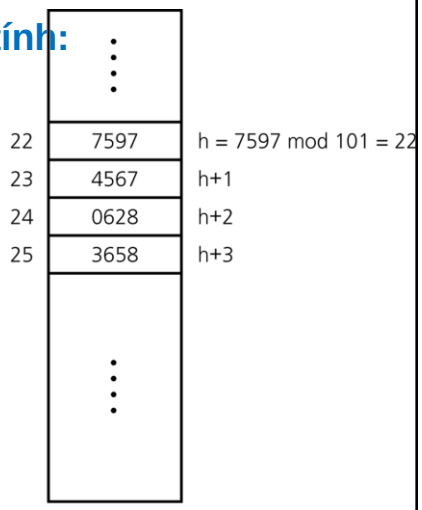
© FIT-HCMUS

Phương pháp dò tuyến tính

19

- **Phương pháp dò tuyến tính:**

$H(k, i) = (h(k) + i) \bmod M$

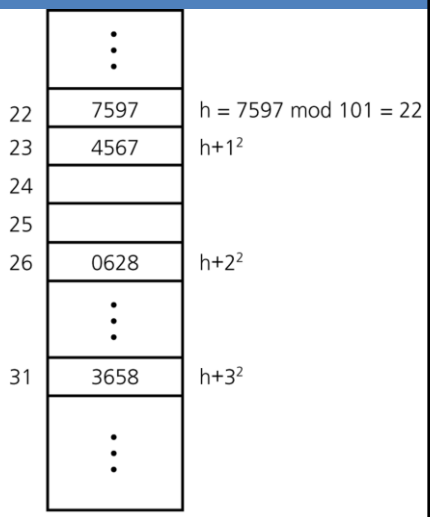


Phương pháp dò bậc 2

20

- ***Phương pháp dò bậc 2:***

$H(k, i) = (h(k) + i^2) \bmod M$

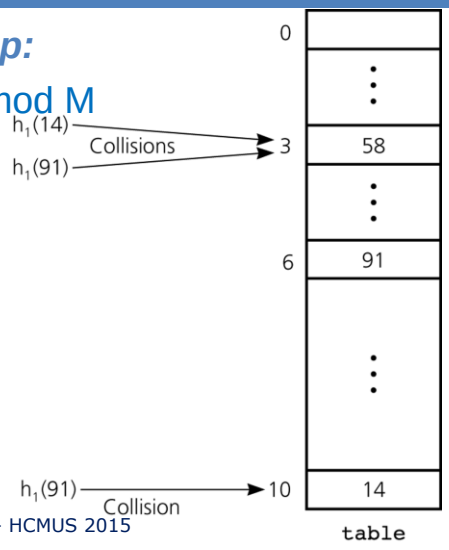


Phương pháp băm kép

21

- ***Phương pháp băm kép:***

$H(k, i) = (h_1(k) + i*h_2(k)) \bmod M$

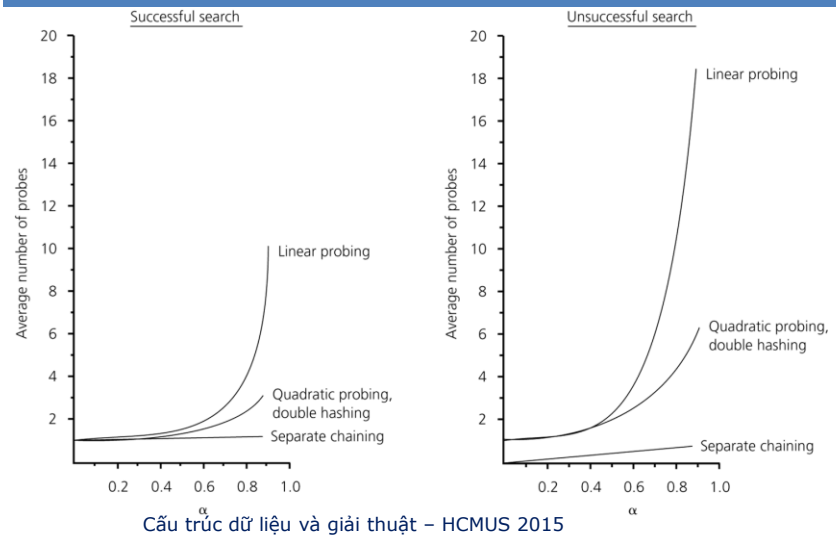


$h_1(key) = key \bmod 11$

$h_2(key) = 7 - (key \bmod 7)$

So sánh các phương pháp giải quyết đụng độ

22



Nhận xét

23

- Phương pháp địa chỉ mở:
  - Đơn giản khi cài đặt.
  - Sử dụng cấu trúc dữ liệu cơ bản.
  - Giải quyết được đụng độ nhưng lại có thể gây ra đụng độ mới.
- Phương pháp nối kết:
  - Không bị ảnh hưởng về tốc độ khi mảng gần đầy.
  - Ít tốn bộ nhớ khi mảng thưa (ít phần tử).

Bài tập

24

1. Cho bảng băm có kích thước M = 11. Hàm băm: h(k) = k mod M. Dùng phương pháp địa chỉ mở. Cho biết kết quả sau khi thêm vào bảng băm các khóa 10, 22, 31, 4, 15, 28, 17, 88, 59, với 3 phương pháp xử lý đụng độ:

   a. Dò tuyến tính.

   b. Dò bậc 2.

   c. Băm kép $h_2(k) = (k \bmod 19)+1$.

© FIT-HCMUS

Bài tập

25

2. Cho từ điển Anh – Việt có 15.000 từ, hãy tổ chức cấu trúc dữ liệu bảng băm và cho biết hàm băm thích hợp giúp cho việc tra từ hiệu quả nhất.

26

Hỏi và Đáp