

Bài 10: Bảng băm

Giảng viên: Hoàng Thị Điệp

Khoa Công nghệ Thông tin – Đại học Công Nghệ

Kiểm tra viết, 15 phút

(Sinh viên có thể sử dụng tài liệu.)

1. Nêu 2 hàm băm
2. Nêu 2 phương pháp giải quyết va chạm trong bảng băm

nói tới trong Chương 9 Giáo trình.

Nội dung chính

- Giới thiệu phương pháp băm
 - Hashing
- Các hàm băm
 - Hash function
- Các chiến lược giải quyết va chạm
 - Collision resolution

KDLTT từ điển

- Trường hợp riêng của tập động khi ta chỉ quan tâm tới tìm kiếm, xen, loại
- Là tập hợp trong đó mỗi phần tử là một cặp (khóa, dữ liệu)
 - Có thể tìm kiếm theo khóa
 - Được sắp hoặc không được sắp
- Các phần tử có thể có cùng khóa*
 - Dictionary vs. Map
- Ứng dụng
 - Từ vựng – nghĩa
 - Tên miền – địa chỉ IP
 - Mã sinh viên – hồ sơ SV
- Các phép toán
 - `find(k)` trả về 1 phần tử có khóa k. Nếu không thấy trả về NULL.
 - `findAll(k)`
 - `insert(k, v)` thêm phần tử (k, v) và trả về con trỏ tới nó
 - `erase(k)` loại bỏ phần tử bất kì có khóa bằng k
 - `erase(p)` loại bỏ phần tử trỏ bởi p
 - `size()` trả về số lượng phần tử
 - `empty()` kiểm tra xem từ điển rỗng hay không

Phương án cài KDLT từ điển

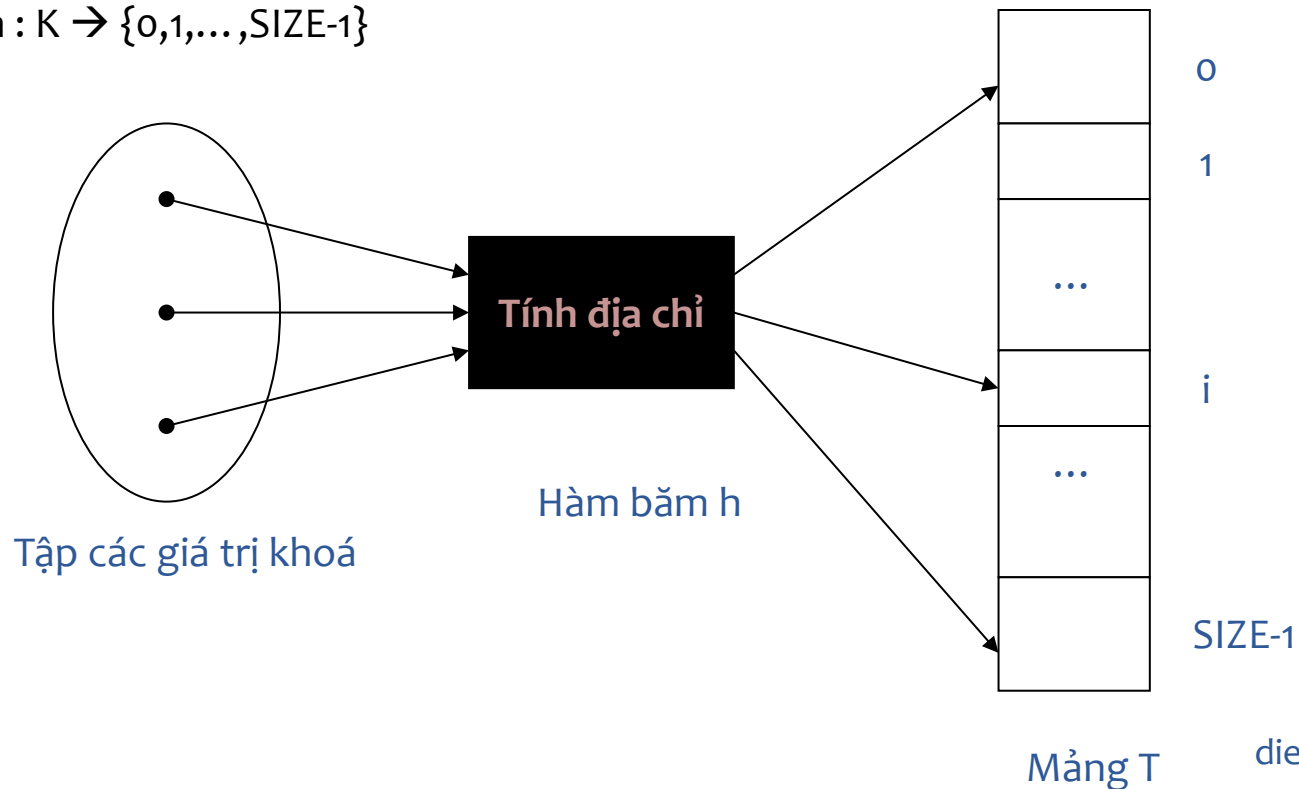
- Mảng được sắp / không được sắp
- DSLK đơn/kép được sắp / không được sắp
- Cây tìm kiếm nhị phân

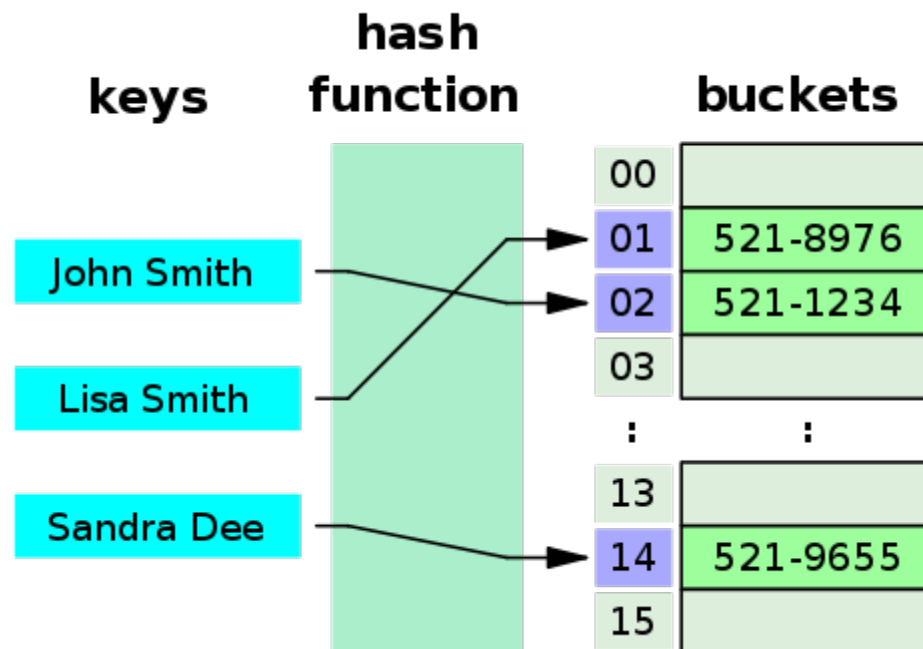
Cài KDUTT từ điển bằng mảng

- Nếu khoá của dữ liệu là số nguyên không âm và nằm trong khoảng $[0..SIZE-1]$
 - có thể sử dụng một mảng data có cỡ SIZE
 - dữ liệu có khoá k sẽ được lưu trong $data[k]$
 - tìm kiếm, xen, loại đều thực hiện trong thời gian $O(1)$
- Thực tế không khả thi vì
 - số phần tử dữ liệu có thể rất nhỏ so với SIZE
 - khoá có thể không phải là số nguyên
- Ta muốn lợi dụng tính ưu việt của phép truy cập trực tiếp của mảng

Phương pháp băm

- Lưu tập dữ liệu trong mảng T với cỡ là SIZE
- Hàm băm: là hàm ứng với mỗi giá trị khoá k của dữ liệu với một chỉ số i ($0 \leq i \leq \text{SIZE}-1$)
 - Dữ liệu này sẽ được lưu trong $T[i]$
 - $h : K \rightarrow \{0, 1, \dots, \text{SIZE}-1\}$





Sự va chạm

- Nếu
 - có $k_1 \neq k_2$ thì $h(k_1) \neq h(k_2)$, và
 - việc tính chỉ số $h(k)$ ứng với mỗi khoá k chỉ đòi hỏi thời gian hằng

thì các phép toán tìm kiếm, xen, loại chỉ cần thời gian $O(1)$

- Va chạm
 - Trong thực tế $k_1 \neq k_2$ có thể cho $h(k_1) = h(k_2)$
- Giải quyết va chạm như thế nào?

Hàm băm

- Hàm băm tốt
 - tính nhanh và dễ dàng
 - đảm bảo ít va chạm
- Một số hàm băm
 - Khóa là số nguyên không âm
 - Phương pháp chia
 - Phương pháp nhân
 - Khóa là xâu ký tự: đổi xâu thành số nguyên không âm

Khóa là số nguyên không âm

- Phương pháp chia
 - $h(k) = k \bmod \text{SIZE}$
 - nhạy cảm với cỡ của bảng băm
 - chọn SIZE để hạn chế xảy ra va chạm
 - số nguyên tố có dạng đặc biệt, chẳng hạn có dạng $4k+3$

- Phương pháp nhân
 - $h(k) = \lfloor (\alpha k - \lfloor \alpha k \rfloor) \cdot \text{SIZE} \rfloor$
 - Ký hiệu $\lfloor x \rfloor$ chỉ phần nguyên của số thực x
 - Thực tế thường chọn

$$\alpha = \Phi^{-1} \approx 0,61803399$$

Khoá là xâu ký tự

- Trước tiên, đổi các xâu ký tự thành các số nguyên, dùng bảng mã ASCII
 - Xâu ký tự có thể xem như một số trong hệ đếm cơ số 128
 - Sau đó chuyển sang hệ đếm cơ số 10
 - Ví dụ
$$\text{"NOTE"} \rightarrow 'N'.128^3 + 'O'.128^2 + 'T'.128 + 'E' =$$
$$= 78.128^3 + 79.128^2 + 84.128 + 69$$
 - Nhược điểm: xâu dài cho kết quả vượt quá khả năng biểu diễn của máy tính
 - Cải tiến: Xâu ký tự thường được tạo thành từ 26 chữ cái và 10 chữ số, và một vài ký tự khác. Thay 128 bởi 37
 - Tính số nguyên ứng với xâu ký tự theo luật Horner
 - Ví dụ
$$\text{"NOTE"} \rightarrow 78.37^3 + 79.37^2 + 84.37 + 69 =$$
$$= ((78.37 + 79).37 + 84).37 + 69$$

Giải quyết va chạm

- Dữ liệu d_1 với khoá k_1 đã được lưu trong $T[i]$, $h(k_1)=i$. Ta cần thêm dữ liệu d_2 với khoá k_2
 - nếu $h(k_2) = i$ thì dữ liệu d_2 cần được đặt vào vị trí nào?
- Các phương pháp
 - Phương pháp định địa chỉ mở (open addressing/probing)
 - mỗi khi xảy ra va chạm, tiến hành thăm dò để tìm một vị trí còn trống trong bảng và đặt dữ liệu mới vào đó
 - Phương pháp tạo dây chuyền (separate chaining)
 - tạo ra một CTDL lưu giữ tất cả các dữ liệu có cùng vị trí i và “gắn” CTDL này vào vị trí đó trong bảng

Phương pháp định địa chỉ mở

- Giả sử vị trí ứng với khoá k là i , $i=h(k)$
 - Từ vị trí này, lần lượt xem xét các vị trí $i_0, i_1, i_2, \dots, i_m, \dots$
 - Trong đó $i_0 = i$, i_m là vị trí thăm dò lần thứ m .
 - Dãy này được gọi là dãy thăm dò.
- Xác định dãy thăm dò
 - Thăm dò tuyến tính (linear probing)
 - Dãy thăm dò là $i, i+1, i+2, \dots$
 - Thăm dò bình phương (quadratic probing)
 - Dãy thăm dò là $i, i + 1^2, i + 2^2, \dots, i + m^2, \dots$
 - Băm kép (double hashing)
 - Dãy thăm dò là $h_1(k) + m h_2(k)$, với $m = 0, 1, 2, \dots$

Các phép toán

- Tìm kiếm? Xen? Loại?
- Minh họa
 - SIZE = 11
 - Thăm dò tuyến tính
 - insert(388) \Rightarrow T[3]
 - insert(130) \Rightarrow T[9]
 - insert(13) \Rightarrow T[2]
 - insert(14) \Rightarrow T[3] \Rightarrow T[4]
 - insert(926) \Rightarrow T[2] \Rightarrow T[3] \Rightarrow T[4] \Rightarrow T[5]

- insert(47) \Rightarrow T[3] \Rightarrow T[4] \Rightarrow T[5] \Rightarrow T[6]
- find(47) ...
- remove(926) ..
- find(47)
- remove(388), find(926)

{không có data,
data bị xóa,
có data}

0	1	2	3	4	5	6	7	8	9	10

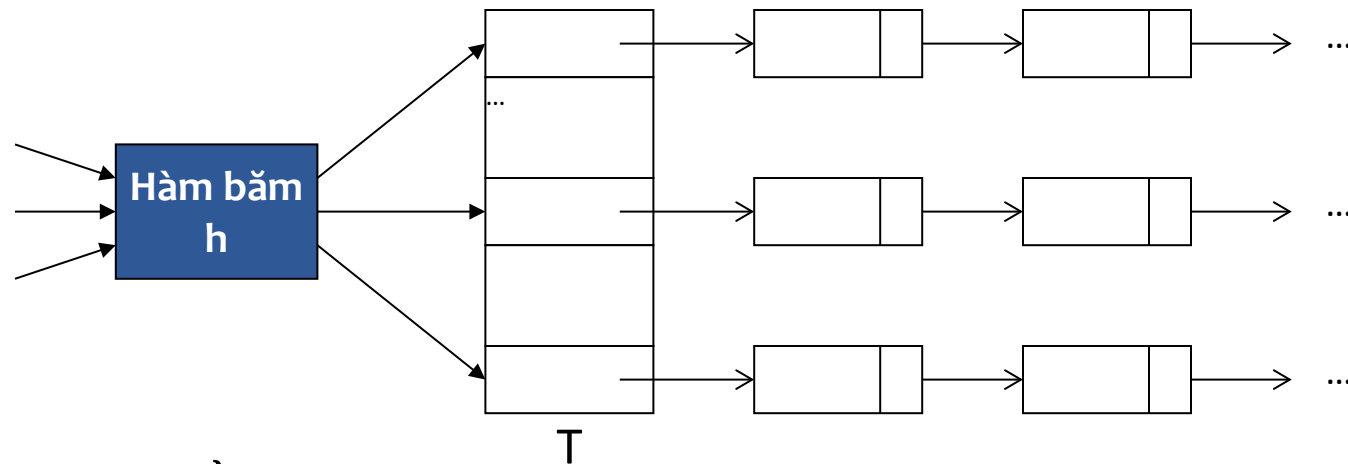
Nhận xét (1/2)

- Thăm dò tuyến tính
 - Ưu điểm: cho phép xét tất cả các vị trí trong mảng
 - phép insert luôn thực hiện được, trừ khi mảng đầy
 - Nhược điểm:
 - dữ liệu tập trung thành các đoạn
 - tìm kiếm tuần tự trong từng đoạn

Nhận xét (2/2)

- Thăm dò bình phương
 - Ưu điểm: tránh được nhược điểm của thăm dò tuyến tính
 - Nhược điểm: không cho phép ta tìm đến tất cả các vị trí trong mảng
 - phép insert có thể không thực hiện được
 - nếu cỡ của mảng là số nguyên tố, thì thăm dò bình phương cho phép ta tìm đến một nửa số vị trí trong mảng
- Băm kép
 - nếu cỡ của mảng và bước thăm dò $h_2(k)$ nguyên tố cùng nhau thì phương pháp băm kép cho phép tìm đến tất cả các vị trí trong mảng

Phương pháp tạo dây chuyền

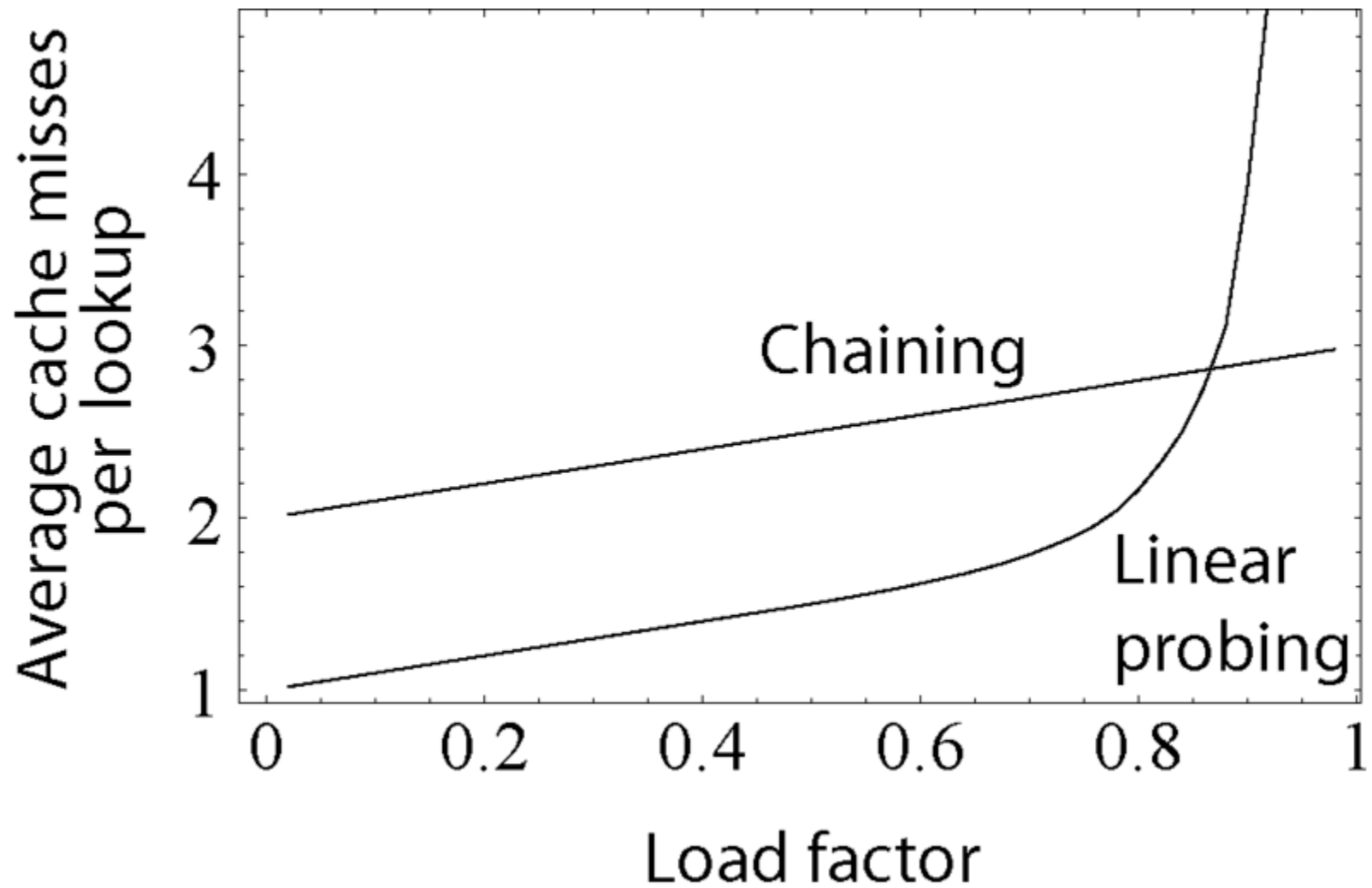


- Ưu điểm
 - số dữ liệu được lưu không phụ thuộc vào cỡ của mảng
- Các phép toán
 - Tìm kiếm?
 - Xen?
 - Loại?

Hiệu quả của phương pháp băm

- Tham số α $\alpha = \frac{N}{SIZE}$
- Băm đ/c mở: mức độ đầy (load factor)
 - α tăng thì khả năng va chạm tăng
 - Khi thiết kế, cần đánh giá max của N để lựa chọn SIZE
 - α không nên vượt quá 2/3
- Băm dây chuyền: độ dài trung bình của một dây chuyền

		Băm đ/c mở, Thăm dò tuyến tính	Băm đ/c mở, Thăm dò bình phương	Băm dây chuyền
Thời gian trung bình	Tìm kiếm thành công	$\frac{1}{2} \left(1 + \frac{1}{1-\alpha} \right)$	$\frac{-\ln(1-\alpha)}{\alpha}$	$1 + \frac{1}{\alpha}$
	Tìm kiếm thất bại	$\frac{1}{2} \left(1 + \frac{1}{(1-\alpha)^2} \right)$	$\frac{1}{1-\alpha}$	α



Chuẩn bị tuần tới

- Lý thuyết: Đọc chương 10 (Hàng ưu tiên)
- Thực hành: Cài đặt bảng băm