BẢNG BĂM (HASH TABLE)

DATA STRUCTURES AND ALGORITHMS

ThS Nguyễn Thị Ngọc Diễm
diemntn@uit.edu.vn

Nội dung

- **Giới thiệu về bảng băm**
- **Hàm băm**
- **Sự đụng độ**
- Các phương pháp giải quyết đụng độ
  - Nối kết trực tiếp - Direct Chaining
  - Nối kết hợp nhất - Coalesced Chaining
  - Dò tuyến tính - Linear probing
  - Dò bậc hai - Quadratic probing
  - Băm kép - Double hashing

Giới thiệu

- Phép băm được đề xuất và hiện thực trên máy tính từ những năm 50 của thế kỷ 20. Dựa trên ý tưởng: chuyển đổi khóa thành một số (xử lý băm) và sử dụng số này để đánh chỉ số cho bảng dữ liệu.
- Các phép toán trên các cấu trúc dữ liệu như danh sách, cây nhị phân, … phần lớn được thực hiện bằng cách so sánh các phần tử của cấu trúc, do vậy thời gian truy xuất không nhanh và phụ thuộc vào kích thước của cấu trúc. Trong khi đó các phép toán trên bảng băm sẽ giúp hạn chế số lần so sánh, và vì vậy sẽ cố gắng giảm thiểu được thời gian truy xuất. Độ phức tạp của các phép toán trên bảng băm thường có bậc là O(1) và không phụ thuộc vào kích thước của bảng băm.

Các thuật ngữ thường dùng

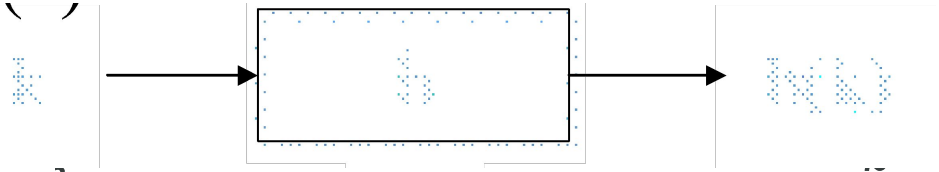
- Hash table (bảng băm)
- Hashing (phép băm)
- Hash function (hàm băm)
- Collision resolution (giải quyết đụng độ)
- Open addressing (Địa chỉ mở)

Các thuật ngữ thường dung (tt)

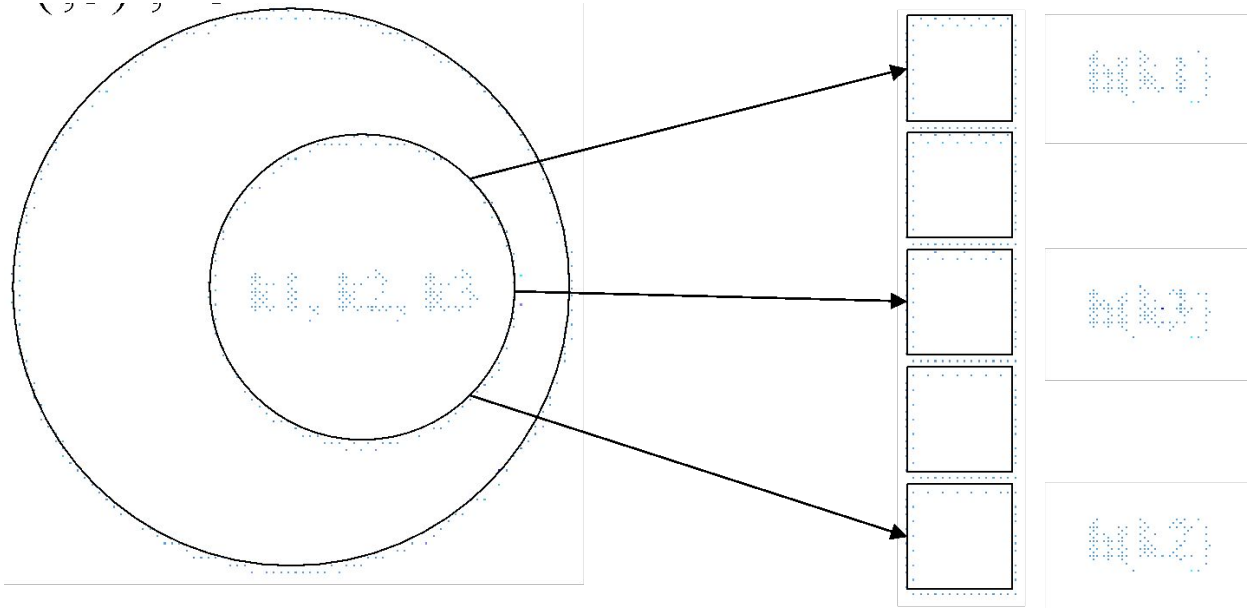
- Phép băm (Hashing): Là quá trình ánh xạ một giá trị khóa vào một vị trí trong bảng.
- Một hàm băm (Hash function) ánh xạ các giá trị khóa đến các vị trí ký hiệu: h hay HF
- Bảng băm (Hash Table) là mảng lưu trữ các record, ký hiệu: HT
- HT có M vị trí được đánh chỉ mục từ 0 đến M-1, M là kích thước của bảng băm.
- Bảng băm thích hợp cho các ứng dụng được cài đặt trên đĩa và bộ nhớ, là một cấu trúc dung hòa giữa thời gian truy xuất và không gian lưu trữ.

Hàm băm (Hash Function)

- Hàm băm biến đổi một khóa vào một bảng các địa chỉ.



- Khóa có thể là dạng số hay số dạng chuỗi.
- Địa chỉ tính ra được là số nguyên trong khoảng 0 đến M-1 với M là số địa chỉ có trên bảng băm

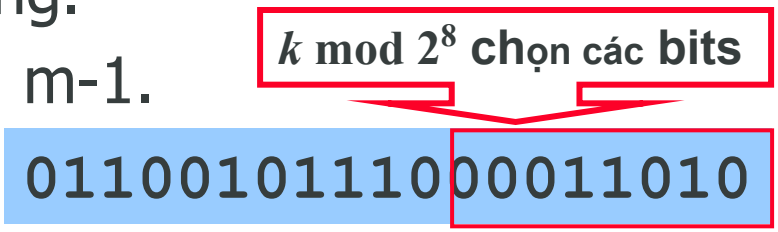


Hàm băm (Hash Function)

- Hàm băm tốt thỏa mãn các điều kiện sau:
  - Tính toán nhanh
  - Các khoá được phân bố đều trong bảng
  - Ít xảy ra đụng độ
  - Xử lý được các loại khóa có kiểu dữ liệu khác nhau
- Giải quyết vấn đề băm với các khoá không phải là số nguyên:
  - Tìm cách biến đổi khóa thành số nguyên.
    - Ví dụ: loại bỏ dấu "-" trong "9635-8904" đưa về số nguyên 96358904
    - Đối với chuỗi, sử dụng giá trị các ký tự trong bảng mã ASCCI
- Sau đó sử dụng các hàm băm chuẩn trên số nguyên.

HF: Phương pháp chia

- Dùng số dư: $h(k) = k \bmod m$
  - k là khoá, m là kích thước của bảng.
  - Như vậy h(k) sẽ nhận: 0, 1, 2, …, m-1.
- **Vấn đề chọn giá trị m**
  - $m = 2^n$: **không tốt** => giá trị của h(k) sẽ là n bits cuối cùng trong biểu diễn nhị phân của k.
  - $m = 10^n$: **không tốt** => giá trị của h(k) sẽ là n chữ số cuối cùng trong biểu diễn thập phân của k.
  - m là nguyên tố: **tốt**
    - Gia tăng sự phân bố đều
    - Thông thường m được chọn là số nguyên tố gần với 2^n
    - Chẳng hạn bảng ~4000 mục, chọn $m = 4093$



HF: Phương pháp nhân

- Sử dụng: $h(k) = \text{floor}(\ m(kA \bmod 1)\ )$
  - k là khóa, m là kích thước bảng, *A* là hằng số với 0 < *A* < 1 và (kA mod 1) là phần thập phân kA.
- *Chọn m và A*
  - Lợi thế của phương pháp nhân là nó hoạt động tốt như nhau với bất kỳ kích thước m. Và thường thì người ta chọn lũy thừa của 2: $m = 2^p$
  - Sự tối ưu trong việc chọn A phụ thuộc vào đặc trưng của dữ liệu. Theo Knuth chọn $A = \frac{1}{2}(\sqrt{5} - 1) \approx 0.6180339887$ được xem là tốt.
- Ví dụ:
  - k = 270589; m = 1000
  - h(k) = 1000 * (270589* 0.6180339887 mod 1)
  - h(k) = 1000 * (167233.1990 mod 1)
  - h(k) = 1000* 0.199 = 199

Phép băm phổ quát

- Việc chọn hàm băm không tốt có thể dẫn đến xác suất đụng độ lớn.
- Giải pháp:
  - Lựa chọn hàm băm h ngẫu nhiên.
  - Chọn hàm băm độc lập với khóa.
  - Khởi tạo một tập các hàm băm H phổ quát và từ đó h được chọn ngẫu nhiên.
- Một họ các hàm băm ***H*** được gọi là phổ quát nếu với hai khóa x ≠ y bất kỳ thì số lượng hàm băm *h*Î***H*** có h(x) = h(y) không vượt quá |***H***|/m.
- Nếu hàm băm *h*Î***H*** thì xác suất xảy ra đụng độ khi sử dụng hàm băm h không vượt quá 1/m

Hàm băm phổ quát

Định lý:

Họ hàm băm $\boldsymbol{H}=\{H_{a,b}(k)=((a*k+b) \bmod p) \bmod m\}$ là phổ quát.

Trong đó:

-a, b là hai số tự nhiên tùy ý, $1 \leq a \leq p-1$, $0 \leq b \leq p-1$

-p là số nguyên tố lớn hơn tất cả các giá trị khóa.

Hàm băm phổ quát

Ví dụ: Xác định hàm băm cho bảng băm T có kích thước 11 và có giá trị khóa lớn nhất là 48.

-Chọn p = 53 là số nguyên tố lớn hơn 48.

-Chọn a = 2 và b = 1.

Hàm băm cần tìm là:

$$h(k) = ((2 * k + 1) \bmod 53) \bmod 11$$

Sự đụng độ (Collision)

- Sự đụng độ là hiện tượng các khóa khác nhau nhưng băm cùng địa chỉ như nhau.
- Khi key1<>key2 mà f(key1)=f(key2) chúng ta nói nút có khóa key1 đụng độ với nút có khóa key2.
- Thực tế người ta giải quyết sự đụng độ theo hai phương pháp: phương pháp nối kết (chaining) và phương pháp băm lại (open adressing).

Quy trình thực hiện lưu trữ bằng bảng băm

Quy trình thực hiện lưu trữ bằng bảng băm được thực hiện qua 2 bước:

- *Bước 1:* Xác định hàm băm để biến đổi khóa cần tìm thành địa chỉ trong bảng băm.
- *Bước 2:* Giải quyết đụng độ (collision) cho trường hợp những khóa khác cho ra cùng một địa chỉ trong bảng băm.

Giải quyết xung đột - Collision Resolution

Gồm các phương pháp:

1. Các loại bảng băm giải quyết sự xung đột bằng phương pháp nối kết như:

- Nối kết trực tiếp - Direct Chaining
- Nối kết hợp nhất - Coalesced Chaining

- Các phần tử bị băm cùng địa chỉ (các phần tử bị xung đột) được gom thành một danh sách liên kết. Lúc này mỗi phần tử trên bảng băm cần khai báo thêm trường liên kết next chỉ phần tử kế bị xung đột cùng địa chỉ.
- Bảng băm giải quyết sự xung đột bằng phương pháp này cho phép tổ chức các phần tử trên bảng băm rất linh hoạt: khi thêm một phần tử vào bảng băm chúng ta sẽ thêm phần tử này vào danh sách liên kết thích hợp phụ thuộc vào băm. Tuy nhiên bảng bảng băm loại này bị hạn chế về tốc độ truy xuất.

Giải quyết xung đột - collision Resolution

2. Các loại bảng băm giải quyết sự xung đột bằng phương pháp băm lại như: (open addressing)

- Dò tuyến tính - Linear probing
- Dò bậc hai - Quadratic probing
- Băm kép - Double hashing

- Nếu băm lần đầu bị xung đột thì băm lại lần 1, nếu bị xung đột nữa thì băm lai lần 2,… Quá trình băm lại diễn ra cho đến khi không còn xung đột nữa. Các phép băm lại (rehash function) thường sẽ chọn địa chỉ khác cho các phần tử.
- Để tăng tốc độ truy xuất, các bảng băm giải quyết sự xung đột bằng phương pháp băm lại thường được cài đặt bằng danh sách kề. Tuy nhiên việc tổ chức các phần tử trên bảng băm không linh hoạt vì các phần tử chỉ được lưu trữ trên một danh sách kề có kích thước đã xác định trước.

Chúc các em học tốt!