



ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

SLIDE BÀI GIẢNG

MÔN

CẤU TRÚC DỮ LIỆU VÀ

GIẢI THUẬT

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN, KHU PHỐ 6, PHƯỜNG LINH TRUNG, QUẬN THỦ ĐỨC, TP. HỒ CHÍ MINH

[T] 08 3725 2002 101 | [F] 08 3725 2148 | [W] www.uit.edu.vn | [E] info@uit.edu.vn



ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

CHƯƠNG V

BẢNG BĂM



MỤC TIÊU CHƯƠNG V

- ❖ Hiểu các khái niệm bảng băm, hàm băm, giải quyết đụng độ
- ❖ Hiểu các thao tác trên bảng băm.
- ❖ Áp dụng bảng băm trong bài toán tìm kiếm
- ❖ Triển khai cấu trúc bảng băm với C++
- ❖ Biết các thuật ngữ tiếng Anh về bảng băm



NỘI DUNG CHƯƠNG V

- I. CÁC KHÁI NIỆM
- II. CÁC DẠNG HÀM BẮM
- III. CÁC PHƯƠNG PHÁP GIẢI QUYẾT ĐỘNG ĐỘ
- IV. SO SÁNH CÁC PHƯƠNG PHÁP KHÁC
- V. DÙNG THƯ VIỆN
- VI. BÀI TẬP



I. CÁC KHÁI NIỆM

❖ ĐẶT VẤN ĐỀ

Chi phí thời gian tìm kiếm:

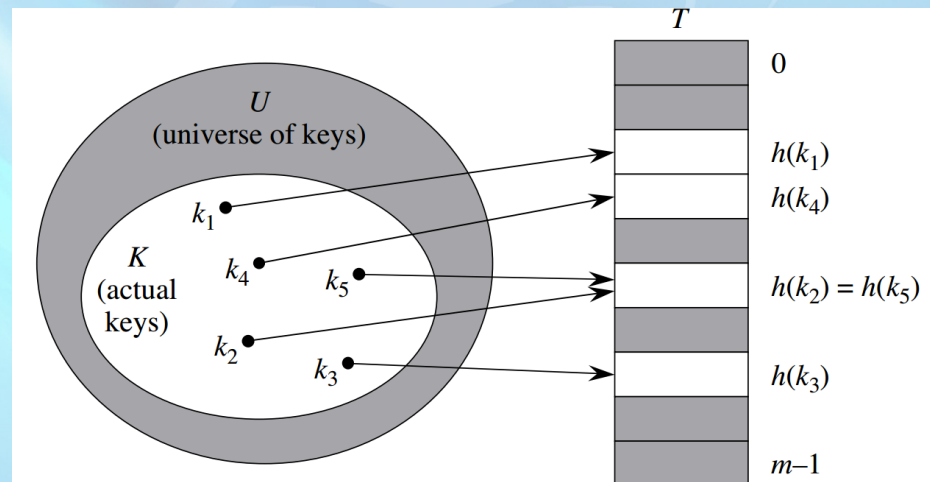
- Cấu trúc mảng, danh sách liên kết + tìm tuyến tính: $O(n)$.
 - Cấu trúc mảng, cây nhị phân + tìm nhị phân: $O(n \log n)$
 - Cấu trúc mảng + tìm nội suy trong điều kiện lý tưởng: $O(1)$
- Xây dựng cấu trúc + thuật toán tìm kiếm với chi phí thời gian đạt $O(1)$



I. CÁC KHÁI NIỆM

❖ Ý TƯỞNG

- Dùng cấu trúc mảng để truy xuất ngẫu nhiên.
- Ánh xạ giá trị cần tìm vào tập chỉ số của mảng
→ Tìm bằng cách truy xuất ngẫu nhiên phần tử của mảng.





I. CÁC KHÁI NIỆM

❖ KHÓA

Khóa (key) là thuộc tính được dùng để xác định một phần tử.

Ví dụ: thông tin sinh viên gồm mã số sinh viên (MSSV), họ tên, địa chỉ. MSSV là khóa theo yêu cầu tìm sinh viên theo mã số.

| MSSV | Họ tên | Địa chỉ |
|----------|------------------|-----------------|
| 19522011 | Lương Trung Kiên | Tp. Hồ Chí Minh |
| 19522110 | Hoàng Kỳ Sơn | Tp. Vinh |

Keys

Two red arrows originate from the word 'Keys' in an orange oval and point to the MSSV values '19522011' and '19522110' in the table.



I. CÁC KHÁI NIỆM

❖ BẢNG BĂM

Bảng băm (hash table) là một cấu trúc dữ liệu có các đặc điểm sau:

- Có kích thước m xác định.
- Cho phép truy xuất ngẫu nhiên từng phần tử theo giá trị khóa.
- Độ phức tạp thời gian có thể đạt $O(1)$ khi truy xuất phần tử.



I. CÁC KHÁI NIỆM

❖ HÀM BĂM

Hàm băm h ánh xạ khóa k của phần tử cần lưu trữ vào vị trí $h(k)$ trong bảng băm có kích thước m :

$$h: \text{KEYS} \rightarrow \{0, 1, \dots, m-1\}$$

Ví dụ: $h : \{19522011, 19522110\} \rightarrow \{0, 1, 2\}$

$$h(k) = k \bmod 3, k \in \{19522011, 19522110\}$$



I. CÁC KHÁI NIỆM

❖ ĐỤNG ĐỘ

Đụng độ (collision) là trường hợp hai giá trị khoá được ánh xạ vào cùng một giá trị chỉ số mảng.

Ví dụ: $h : \{19522011, 19522110\} \rightarrow \{0, 1, 2\}$

$$h(k) = k \bmod 3, k \in \{19522011, 19522110\}$$

Có $h(19522011) = 0$ và $h(19522110) = 0$

- Thiết kế hàm băm để giảm đụng độ
- Xử lý đụng độ



I. CÁC KHÁI NIỆM

❖ HỆ SỐ TẢI

Hệ số tải (load factor) của bảng băm T có kích thước m khi lưu trữ n phần tử là $\lambda = n/m$

Ví dụ: hệ số tải của bảng băm T bên dưới là $\lambda = 0.57$

| | |
|---|---|
| 0 | x |
| 1 | |
| 2 | y |
| 3 | z |
| 4 | |
| 5 | |
| 6 | t |



NỘI DUNG CHƯƠNG V

- I. CÁC KHÁI NIỆM
- II. CÁC DẠNG HÀM BẮM
- III. CÁC PHƯƠNG PHÁP GIẢI QUYẾT ĐỘ ĐỘ
- IV. SO SÁNH CÁC PHƯƠNG PHÁP KHÁC
- V. DÙNG THƯ VIỆN
- VI. BÀI TẬP



II. CÁC DẠNG HÀM BẮM

❖ YÊU CẦU ĐỐI VỚI HÀM BẮM

- Biến đổi giá trị khóa từ kiểu dữ liệu bất kỳ thành số tự nhiên.
- Khả năng mỗi khoá được băm (hash) vào vị trí bất kì trong bảng băm là tương đương, không phụ thuộc vào các khoá đã được băm trước đó.



II. CÁC DẠNG HÀM BẮM

❖ HÀM BẮM DÙNG PHÉP CHIA

Dạng của hàm băm dùng phép chia:

$$h(k) = k \bmod m$$

Trong đó: k , m là và kích thước của bảng băm.

Lưu ý khi chọn giá trị cho m :

- m không nên là lũy thừa của 2.
- m nên là 1 số nguyên tố không gần với một số là lũy thừa của 2

VD: Bảng băm có 4000 mục, chọn $m = 4093$



II. CÁC DẠNG HÀM BẮM

❖ HÀM BẮM DÙNG PHÉP CHIA (tt)

VD: Cho bảng băm T có kích thước $m=11$, khóa là mã hàng gồm 3 ký tự ASCII. Hãy xây dựng hàm băm dùng phép chia cho T và tính chỉ số cho các mã hàng “ATK”, “SOS” và “TNT”.



II. CÁC DẠNG HÀM BẮM

❖ HÀM BẮM DÙNG PHÉP CHIA (tt)

- Biến đổi chuỗi ký tự thành số:

$$\text{toInt}(s) = s[0] + s[1] * 37 + s[2] * 37^2$$

- Hàm băm: $h(s) = \text{toInt}(s) \bmod 11$

Khi đó:

- $h(\text{"ATK"}) = (65 + 84 * 37 + 75 * 37^2) \bmod 11 = 6$
- $h(\text{"SOS"}) = (83 + 79 * 37 + 83 * 37^2) \bmod 11 = 0$
- $h(\text{"TNT"}) = (84 + 78 * 37 + 84 * 37^2) \bmod 11 = 2$



II. CÁC DẠNG HÀM BẮM

❖ HÀM BẮM DÙNG PHÉP NHÂN

Dạng của hàm băm dùng phép nhân:

$$h(k) = \lfloor m * (k * A \bmod 1) \rfloor$$

Trong đó:

- m , k là kích thước bảng băm và giá trị khóa.
- $A \in (0, 1)$ là hằng số.
- $X \bmod 1$ là lấy phần thập phân của X
- $\lfloor X \rfloor$ là lấy giá trị nguyên lớn nhất mà nhỏ hơn X .



II. CÁC DẠNG HÀM BẮM

❖ HÀM BẮM DÙNG PHÉP NHÂN (tt)

Lưu ý:

- m nên chọn là một số có dạng 2^p , $p \in \mathbb{N}^*$
- Theo Knuth, $A = (\text{sqrt}(5) - 1) / 2$.



II. CÁC DẠNG HÀM BẮM

❖ HÀM BẮM PHỔ QUÁT

Một họ các hàm băm H được gọi là phổ quát nếu với hai khóa $x \neq y$ bất kỳ thì số lượng hàm băm $h \in H$ có $h(x) = h(y)$ không vượt quá $|H|/m$.

Nếu hàm băm $h \in H$ thì xác suất xảy ra đụng độ khi sử dụng hàm băm h không vượt quá $1/m$.



II. CÁC DẠNG HÀM BẮM

❖ HÀM BẮM PHỔ QUÁT (tt)

Định lý:

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



II. CÁC DẠNG HÀM BẮM

❖ HÀM BẮM PHỔ QUÁT (tt)

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:

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



NỘI DUNG CHƯƠNG V

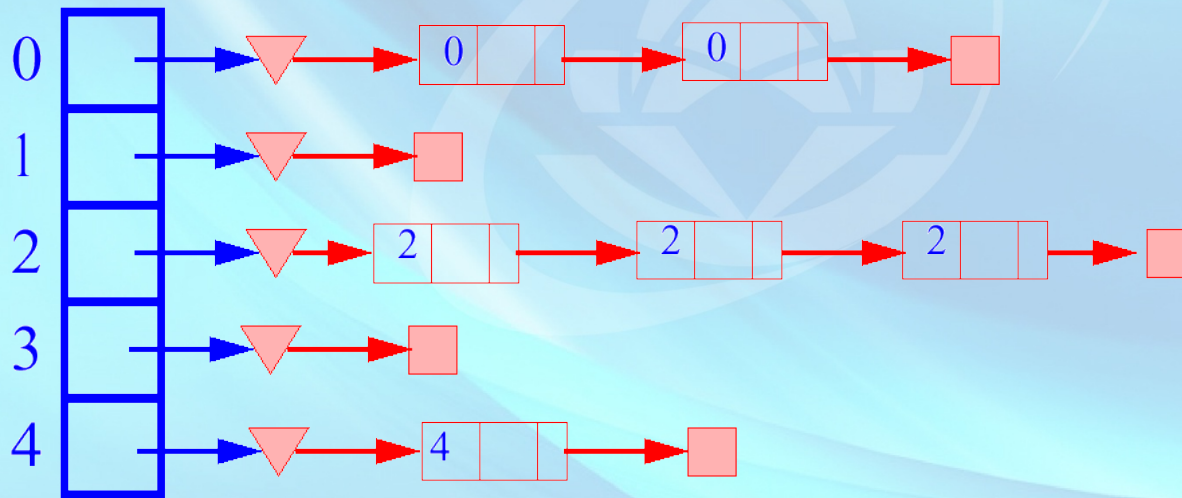
- I. CÁC KHÁI NIỆM
- II. CÁC DẠNG HÀM BẮM
- III. CÁC PHƯƠNG PHÁP GIẢI QUYẾT ĐỘNG ĐỘ**
- IV. SO SÁNH CÁC PHƯƠNG PHÁP KHÁC
- V. DÙNG THƯ VIỆN
- VI. BÀI TẬP



III. GIẢI QUYẾT ĐỤNG ĐỘ

❖ PHƯƠNG PHÁP NỐI KẾT

Phương pháp nối kết (chaining): các phần tử đụng độ tại giá trị băm i được kết nối với nhau thành một danh sách liên kết thứ i trong bảng băm T .





III. GIẢI QUYẾT ĐỤNG ĐỘ

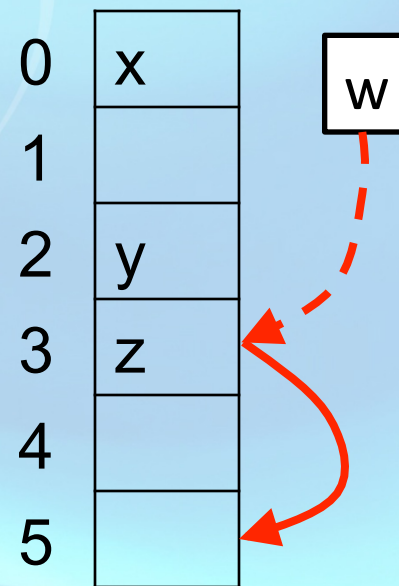
❖ PHƯƠNG PHÁP ĐỊA CHỈ MỞ

Phương pháp địa chỉ mở (open address): nếu giá trị băm i trên bảng băm T đã tồn tại phần tử khác thì thăm dò vị trí phù hợp.

Các vị trí thăm dò được xác định:

$$h_i(k) = (h(k) + f(i)) \bmod m$$

Trong đó $f(0) = 0$.





III. GIẢI QUYẾT ĐỤNG ĐỘ

❖ PHƯƠNG PHÁP ĐỊA CHỈ MỞ (tt)

Thăm dò tuyến tính (linear probing):

Hàm $f(i)$ có dạng $f(i) = a * i$

Ví dụ: Cho biết kết quả thêm lần lượt các phần tử có giá trị khóa 89, 18, 49, 58, 69 vào bảng băm T có kích thước $m = 10$.

Biết hàm băm: $h(k) = (k \bmod 10)$
và hàm giải quyết đụng độ: $f(i) = i$



III. GIẢI QUYẾT ĐỤNG ĐỘ

❖ PHƯƠNG PHÁP ĐỊA CHỈ MỞ (tt)

Thăm dò tuyến tính (linear probing):

→ Hàm băm lại:

$$\begin{aligned}h_i(k) &= (h(k) + f(i)) \bmod 10 \\ &= ((k \bmod 10) + i) \bmod 10\end{aligned}$$



III. GIẢI QUYẾT ĐỤNG ĐỘ

❖ PHƯƠNG PHÁP ĐỊA CHỈ MỞ (tt)

Thăm dò tuyến tính (linear probing):

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---------|----|----|----|---|---|---|---|---|----|----|
| Thêm 89 | | | | | | | | | | 89 |
| Thêm 18 | | | | | | | | | 18 | 89 |
| Thêm 49 | 49 | | | | | | | | 18 | 89 |
| Thêm 58 | 49 | 58 | | | | | | | 18 | 89 |
| Thêm 69 | 49 | 58 | 69 | | | | | | 18 | 89 |



III. GIẢI QUYẾT ĐỤNG ĐỘ

❖ PHƯƠNG PHÁP ĐỊA CHỈ MỞ (tt)

Thăm dò bậc hai (quadratic probing):

Hàm $f(i)$ có dạng $f(i) = a \cdot i^2 + b \cdot i$

Ví dụ: Cho biết kết quả thêm lần lượt các phần tử có giá trị khóa 89, 18, 49, 58, 69 vào bảng băm T có kích thước $m = 10$.

Biết hàm băm: $h(k) = (k \bmod 10)$
và hàm giải quyết đụng độ: $f(i) = i^2$



III. GIẢI QUYẾT ĐỤNG ĐỘ

❖ PHƯƠNG PHÁP ĐỊA CHỈ MỞ (tt)

Thăm dò bậc hai (quadratic probing):

→ Hàm băm lại:

$$\begin{aligned}h_i(k) &= (h(k) + f(i)) \bmod 10 \\ &= ((k \bmod 10) + i^2) \bmod 10\end{aligned}$$



III. GIẢI QUYẾT ĐỤNG ĐỘ

❖ PHƯƠNG PHÁP ĐỊA CHỈ MỞ (tt)

Thăm dò bậc hai (quadratic probing):

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---------|----|---|----|----|---|---|---|---|----|----|
| Thêm 89 | | | | | | | | | | 89 |
| Thêm 18 | | | | | | | | | 18 | 89 |
| Thêm 49 | 49 | | | | | | | | 18 | 89 |
| Thêm 58 | 49 | | 58 | | | | | | 18 | 89 |
| Thêm 69 | 49 | | 58 | 69 | | | | | 18 | 89 |



III. GIẢI QUYẾT ĐỤNG ĐỘ

❖ PHƯƠNG PHÁP ĐỊA CHỈ MỞ (tt)

Băm kép (double hashing):

Hàm $f(i)$ có dạng $f(i) = i * h'(k)$

$h'(k)$ là hàm băm phụ, có dạng:

$$h'(k) = R - (k \bmod R)$$

Trong đó:

- R là số nguyên tố nhỏ hơn m .
- $h'(k)$ không được bằng 0.



III. GIẢI QUYẾT ĐỤNG ĐỘ

❖ PHƯƠNG PHÁP ĐỊA CHỈ MỞ (tt)

Băm kép (double hashing):

Ví dụ: Cho biết kết quả thêm lần lượt các phần tử có giá trị khóa 89, 18, 49, 58, 69 vào bảng băm T có kích thước $m = 10$.

Biết hàm băm: $h(k) = (k \bmod 10)$

và hàm giải quyết đụng độ:

$$f(i) = i * (7 - (k \bmod 7))$$



III. GIẢI QUYẾT ĐỤNG ĐỘ

❖ PHƯƠNG PHÁP ĐỊA CHỈ MỞ (tt)

Băm kép (double hashing):

→ Hàm băm lại:

$$\begin{aligned} h_i(k) &= (h(k) + f(i)) \bmod 10 \\ &= ((k \bmod 10) + i * (7 - (k \bmod 7))) \bmod 10 \end{aligned}$$



III. GIẢI QUYẾT ĐỤNG ĐỘ

❖ PHƯƠNG PHÁP ĐỊA CHỈ MỞ (tt)

Băm kép (double hashing):

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---------|----|---|---|----|---|---|----|---|----|----|
| Thêm 89 | | | | | | | | | | 89 |
| Thêm 18 | | | | | | | | | 18 | 89 |
| Thêm 49 | | | | | | | 49 | | 18 | 89 |
| Thêm 58 | | | | 58 | | | 49 | | 18 | 89 |
| Thêm 69 | 69 | | | 58 | | | 49 | | 18 | 89 |



III. GIẢI QUYẾT ĐỤNG ĐỘ

❖ PHƯƠNG PHÁP ĐỊA CHỈ MỞ (tt)

Áp dụng vào thao tác thêm phần tử.

Đầu vào:

- Bảng băm T có kích thước m,
- Hàm băm lại $h_i(k)$.
- Phần tử có khóa k cần thêm

Đầu ra:

- Bảng băm T.
- Vị trí của phần tử được chèn hoặc -1 nếu gặp lỗi.

Ký hiệu DELETED và NIL được dùng để đánh dấu vị trí đã được xóa và vị trí trống



III. GIẢI QUYẾT ĐỤNG ĐỘ

❖ PHƯƠNG PHÁP ĐỊA CHỈ MỞ (tt)

Áp dụng vào thao tác thêm phần tử.

HASH-INSERT (T, k)

```
1.    i = 0
2.    do
3.        j = h(k, i)
4.        if T[j] == NIL or T[j] == DELETED
5.            T[j] = k
6.            return j
7.        else i = i + 1
8.    while i < m
9.    return -1
```



III. GIẢI QUYẾT ĐỤNG ĐỘ

❖ PHƯƠNG PHÁP ĐỊA CHỈ MỞ (tt)

Áp dụng vào thao tác xóa phần tử.

Đầu vào:

- Bảng băm T có kích thước m,
- Hàm băm lại $h_i(k)$.
- Khóa k cần xóa.

Đầu ra:

- Bảng băm T.
- Vị trí của phần tử được xóa hoặc -1 nếu gặp lỗi.

Ký hiệu DELETED và NIL được dùng để đánh dấu vị trí đã được xóa và vị trí trống



III. GIẢI QUYẾT ĐỤNG ĐỘ

❖ PHƯƠNG PHÁP ĐỊA CHỈ MỞ (tt)

Áp dụng vào thao tác xóa phần tử.

HASH-DELETE (T, k)

```
1.   i = 0
2.   do
3.       j = h(k, i)
4.       if T[j] == k
5.           temp = T[j]
6.           T[j] = DELETED
7.       return temp
8.       else i = i + 1
9.   while T[j] != NIL and i < m
10.  return -1
```




III. GIẢI QUYẾT ĐỤNG ĐỘ

❖ PHƯƠNG PHÁP ĐỊA CHỈ MỞ (tt)

Áp dụng vào thao tác tìm phần tử:

Đầu vào:

- Bảng băm T có kích thước m ,
- Hàm băm lại $h_i(k)$.
- Khóa k cần tìm.

Đầu ra:

- Vị trí của phần tử được tìm hoặc -1 nếu gặp lỗi.

Ký hiệu DELETED và NIL được dùng để đánh dấu vị trí đã được xóa và vị trí trống



III. GIẢI QUYẾT ĐỤNG ĐỘ

❖ PHƯƠNG PHÁP ĐỊA CHỈ MỞ (tt)

Áp dụng vào thao tác tìm phần tử:

HASH-SEARCH (T, k)

```
1.    i = 0
2.    do
3.        j = h(k, i)
4.        if T[j] == k
5.            return j
6.        else i = i + 1
7.    while T[j] != NIL and i < m
8.    return -1
```



III. GIẢI QUYẾT ĐỤNG ĐỘ

❖ BĂM LẠI

Băm lại (rehashing) cần được thực hiện khi:

- Không thể thực hiện một thao tác thêm phần tử mới bất kỳ.
- Bảng băm có hệ số tải đạt một giá trị xác định ($\lambda \geq 0.5$) làm độ phức tạp thời gian của thao tác tìm kiếm tăng.



III. GIẢI QUYẾT ĐỤNG ĐỘ

❖ BẮM LẠI (tt)

Các bước thực hiện khi bắt lại:

- Tăng kích thước bảng bắt lên m' với m' là một số nguyên tố và $m' \approx 2 * m$.
- Cập nhật hàm bắt.
- Bắt lại các phần tử có trong bảng bắt cũ và thêm vào bảng bắt mới.



III. GIẢI QUYẾT ĐỤNG ĐỘ

❖ BẮM LẠI (tt)

Ví dụ: Thêm phần tử có khóa 23 vào bảng băm sau:

| 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|----|---|----|---|---|----|
| 6 | 15 | | 24 | | | 13 |



III. GIẢI QUYẾT ĐỤNG ĐỘ

❖ BẮM LẠI (tt)

Hệ số tải $\lambda = 4/7 > 0.5$

→ Băm lại:

- Tăng kích thước bảng băm lên $m' = 17$
- Hàm băm mới: $h'(k) = k \bmod 17$
- Băm lại các giá trị 6, 15, 24, 13 vào bảng băm mới, sử dụng hàm băm mới
- Băm phần tử 23 vào bảng băm mới.



III. GIẢI QUYẾT ĐỤNG ĐỘ

❖ BẮM LẠI (tt)

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|----|----|---|----|----|----|----|----|----|----|
| | | | | | | 6 | 23 | 24 | | | | | 13 | | 15 | |



NỘI DUNG CHƯƠNG V

- I. CÁC KHÁI NIỆM**
- II. CÁC DẠNG HÀM BẮM**
- III. CÁC PHƯƠNG PHÁP GIẢI QUYẾT ĐỘ ĐỘ**
- IV. SO SÁNH CÁC PHƯƠNG PHÁP KHÁC**
- V. DÙNG THƯ VIỆN**
- VI. BÀI TẬP**



IV. SO SÁNH

| Phương pháp | Xấu nhất | Trung bình | Tốt nhất |
|-----------------------|-------------|-------------|----------|
| Mảng + Tìm tuyến tính | $O(n)$ | $O(n)$ | $O(1)$ |
| Mảng + Tìm nhị phân | $O(\log n)$ | $O(\log n)$ | $O(1)$ |
| Danh sách liên kết | $O(n)$ | $O(n)$ | $O(1)$ |
| Cây nhị phân tìm kiếm | $O(n)$ | $O(\log n)$ | $O(1)$ |
| Cây đỏ đen / cây AVL | $O(\log n)$ | $O(\log n)$ | $O(1)$ |
| Bảng băm | $O(n)$ | $O(1)$ | $O(1)$ |



NỘI DUNG CHƯƠNG V

- I. CÁC KHÁI NIỆM
- II. CÁC DẠNG HÀM BẮM
- III. CÁC PHƯƠNG PHÁP GIẢI QUYẾT ĐỘ ĐỘ
- IV. SO SÁNH CÁC PHƯƠNG PHÁP KHÁC
- V. DÙNG THƯ VIỆN**
- VI. BÀI TẬP



V. DÙNG THƯ VIỆN

❖ UNORDERED_MAP

- `#include <unordered_map>`
- `unordered_map` là một tập hợp các phần tử được truy xuất như bảng băm theo thuộc tính khóa.
- Mỗi phần tử gồm một cặp `key, value` được đặt trong kiểu `pair<key, value>`



V. DÙNG THƯ VIỆN

❖ UNORDERED_MAP

- Mỗi biến kiểu pair có trường pair.first và pair.second lần lượt là trường key và value của một phần tử.
- Các thao tác cơ bản của unordered_map gồm: insert – thêm phần tử, find – tìm phần tử có khóa k, erase – xóa phần tử.

Sinh viên tìm hiểu thêm về unordered_map



V. DÙNG THƯ VIỆN

❖ UNORDERED_MAP

Ví dụ: Viết chương trình nhập vào thông tin sinh viên gồm mã số sinh viên và tên (không chứa khoảng trắng); cho phép tìm thông tin theo mã số sinh viên, in ra màn hình và xóa thông tin sinh viên vừa tìm được.



V. DÙNG THƯ VIỆN

```
#include <iostream>
#include <string>
#include <unordered_map>
using namespace std;
unordered_map<int, string> nhap();
void tim(unordered_map<int, string> &, int);
int main() {
    unordered_map<int, string> ds = nhap();
    int ms;
    cin >> ms;
    tim(ds, ms);
    return 0;
}
```



V. DÙNG THƯ VIỆN

```
unordered_map<int, string> nhap() {  
    unordered_map<int, string> ds;  
    pair<int, string> hs;  
    int n;  
    cin >> n;  
    for (int i = 0; i < n; i++) {  
        cin >> hs.first >> hs.second;  
        ds.insert(hs);  
    }  
    return ds;  
}
```




V. DÙNG THƯ VIỆN

```
void tim(unordered_map<int, string> &ds, int ms){
    unordered_map<int, string>::iterator
        iter = ds.find(ms);
    if (iter != ds.end()) {
        cout << "MSSV: " << (*iter).first
            << "Ten: " << (*iter).second
            << '\n';
        ds.erase(iter);
    }
    else {
        cout << "Khong tim thay\n";
    }
}
```



V. DÙNG THƯ VIỆN

❖ UNORDERED_SET

- `#include <unordered_set>`
- `unordered_set` là một tập hợp các phần tử được truy xuất như bảng băm.
- Mỗi phần tử là một biến có kiểu dữ liệu bất kỳ



V. DÙNG THƯ VIỆN

❖ UNORDERED_MAP

- Các thao tác cơ bản của `unordered_set` gồm: `insert` – thêm phần tử, `find` – tìm phần tử, `erase` – xóa phần tử.

Sinh viên tìm hiểu thêm về `unordered_set`



V. DÙNG THƯ VIỆN

❖ UNORDERED_SET

Ví dụ: Viết chương trình nhập vào một dãy số nguyên; cho phép tìm một số nguyên k trong dãy đã nhập, in ra màn hình “YES” nếu tìm thấy và “NO” nếu không tìm thấy số cần tìm, sau đó xóa số vừa tìm được.



V. DÙNG THƯ VIỆN

```
#include <iostream>
#include <string>
#include <unordered_set>
using namespace std;
unordered_map<int> nhap();
void tim(unordered_set<int> &, int);
int main() {
    unordered_set<int> ds = nhap();
    int k;
    cin >> k;
    tim(ds, k);
    return 0;
}
```



V. DÙNG THƯ VIỆN

```
unordered_set<int> nhap() {  
    unordered_set<int> ds;  
    int n, tmp;  
    cin >> n;  
    for (int i = 0; i < n; i++) {  
        cin >> tmp;  
        ds.insert(tmp);  
    }  
    return ds;  
}
```




V. DÙNG THƯ VIỆN

```
void tim(unordered_set<int> &ds, int k){  
    unordered_set<int>::iterator iter=ds.find(k);  
    if (iter != ds.end()) {  
        cout << "YES\n";  
        ds.erase(iter);  
    }  
    else {  
        cout << "NO\n";  
    }  
}
```



NỘI DUNG CHƯƠNG V

- I. CÁC KHÁI NIỆM
- II. CÁC DẠNG HÀM BẮM
- III. CÁC PHƯƠNG PHÁP GIẢI QUYẾT ĐỘ ĐỘ
- IV. SO SÁNH CÁC PHƯƠNG PHÁP KHÁC
- V. DÙNG THƯ VIỆN
- VI. BÀI TẬP**



VI. BÀI TẬP

- Bài 1: Hãy mô tả các bước xảy ra khi chèn các khoá 5, 28, 19, 15, 20, 33, 12, 17, 10 vào một bảng băm được giải quyết đụng độ bằng phương pháp nối kết. Cho bảng băm có 9 ô và hàm băm là $h(k) = k \bmod 9$



VI. BÀI TẬP

- Bài 2: Xét một bảng băm có kích thước là $m = 1000$ và hàm băm là $h(k) = \lfloor m * (k * A \bmod 1) \rfloor$ với $A = (\sqrt{5} - 1)/2$. Hãy tính vị trí của các khoá 61, 62, 63, 65 khi được ánh xạ.



VI. BÀI TẬP

- Bài 3: Thêm các khoá 10, 22, 31, 4, 15, 28, 17, 88, 59 vào một bảng băm có kích thước $m = 11$ sử dụng địa chỉ mở với hàm băm $h(k) = k \bmod 11$.
Hãy minh họa kết quả khi thêm các khóa này vào bảng băm sử dụng phương pháp dò tuyến tính, dò bậc hai, và băm kép với hàm băm phụ là $h'(k) = 1 + (k \bmod (m-1))$