

# Cấu trúc dữ liệu và giải thuật

## CÁC CHIẾN LƯỢC TÌM KIẾM

Văn Chí Nam – Nguyễn Thị Hồng Nhung – Đặng Nguyễn Đức Tiến  
Vũ Thanh Hưng

# Nội dung trình bày

2

Giới thiệu



```
graph TD; A[Giới thiệu] --> B[Tìm kiếm tuần tự]; B --> C[Tìm kiếm nhị phân]; C --> D[Tìm kiếm theo bảng băm]; D --> E[Tổng kết];
```

Tìm kiếm tuần tự

Tìm kiếm nhị phân

Tìm kiếm theo bảng băm

Tổng kết

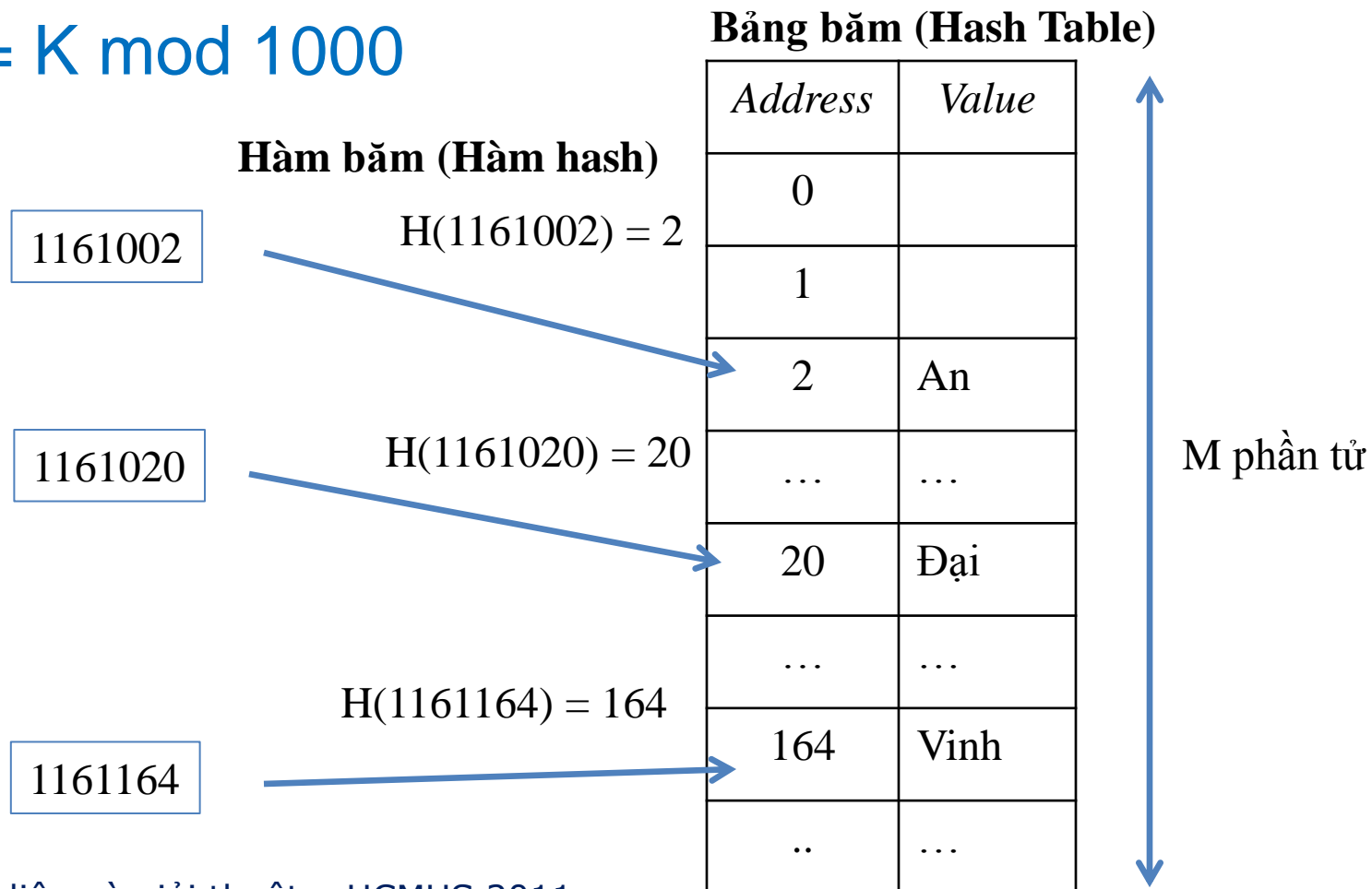
# Có cách nào để chi phí tìm kiếm là $O(1)$ ?

1	1161002	An	17	1161044	HỒ	33	1161116	Tâm
2	1161008	Trâm Anh	18	1161047	Hòa	34	1161123	Thái
3	1161009	Bách	19	1161052	Hoàng	35	1161125	Thành
4	1161010	Bảo	20	1161055	Hùng	36	1161141	Tinh
5	1161011	Bình	21	1161061	Huy	37	1161145	Trang
6	1161012	Bình	22	1161063	Khải	38	1161147	Triết
7	1161015	Châu	23	1161065	Khánh	39	1161149	Trưởng
8	1161020	Đại	24	1161071	Kiên	40	1161150	Trọng
9	1161023	Đăng	25	1161079	Lực	41	1161154	Tự
10	1161028	Định	26	1161086	Nam	42	1161156	Tùng
11	1161031	Đức	27	1161088	Năng	43	1161159	Tuyên
12	1161032	Dung	28	1161089	Nghi	44	1161164	Vinh
13	1161033	Gon	29	1161096	Nhiên	45	1161167	Vũ
14	1161034	Duy	30	1161098	Phong	46	1161169	Vương
15	1161037	Hà	31	1161102	Phúc	47	1161171	Tú
16	1161040	Hiền	32	1161107	Quang	48	1161173	Nhân
						49	1161174	Thám
						50	1161176	Tiên

# Khái quát về hash

4

- ⊙ Biến đổi khóa K (MSSV) thành địa chỉ mảng:
- ⊙  $H(K) = K \bmod 1000$



# Tìm kiếm theo bảng băm

Hash Table

# Khái quát về hash

6

- ⊙ **Vấn đề:** Cho trước 1 tập  $S$  gồm các phần tử được đặc trưng bởi giá trị khóa  $K$ . 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.

# Hàm băm (hash function)

7

- ◉ **Định nghĩa:** là hàm biến đổi khóa  $k$  của phần tử thành địa chỉ trong *bảng băm*.

Ví dụ:  $H(\text{"VCNam"}) = 406$

$H(\text{"NTHHung"}) = 1000$

$H(\text{"DNDTien"}) = 2$

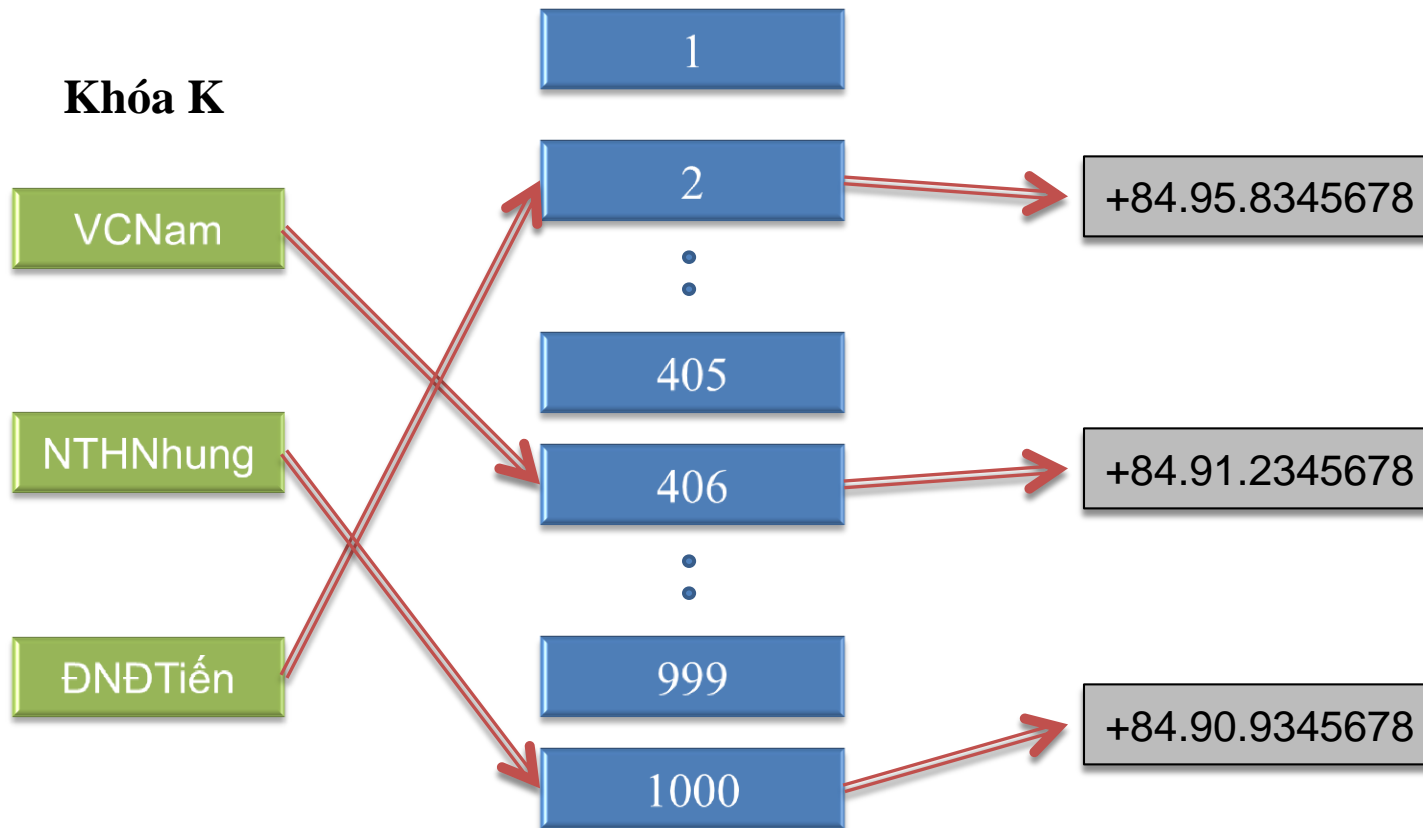
- ◉ **Tổng quát về phép biến đổi khóa:** Là 1 ánh xạ thích hợp từ tập các khóa  $U$  vào tập các địa chỉ  $A$ .

$H: U \rightarrow A$

$k \rightarrow a = h(k)$

# Ví dụ về một bảng băm không nguyên

8



- ◉ Khi các khóa ở dạng chuỗi → chuyển thành mã ASCII và băm nguyên bình thường



# Độ phức tạp

9

- ◉ Chi phí tìm kiếm trung bình:  $O(1)$

# Khó khăn của hàm băm

10

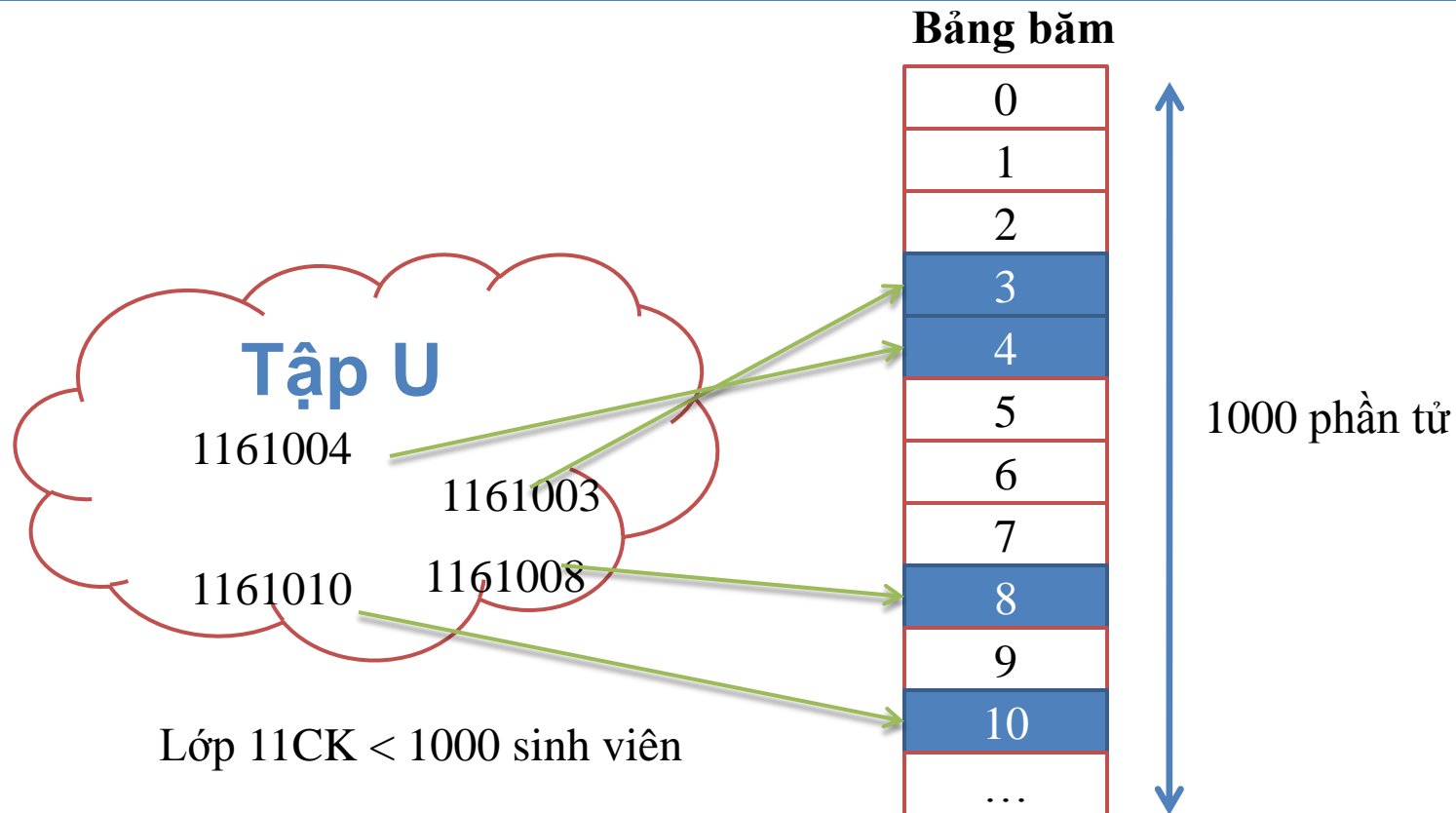
- ⊙ Tập các giá trị khóa (U) có thể lớn hơn rất nhiều so với số khóa thực tế (K) rất nhiều.
- ⊙ **Ví dụ:** Khoa CNTT quyết định sử dụng cấu trúc hash table để quản lý không chỉ 11CK mà cả 10CK cũng với hàm hash:

$$H(K) = K \bmod 1000$$

Chuyện gì sẽ xảy ra?

# Nếu $U = \{11CK\}$

11



# Sự đụng độ (collision)

12

◉  $\exists k_1, k_2 \in K$ :

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

**Tập U**

1061003

1161003

**Bảng băm**

0
1
2
3
4
5
6
7
8
9
10
...

1000 phần tử

**Giải pháp?**

Lớp 11CK, 10CK < 1000 sinh viên

# Các phương pháp xử lý đụng độ

13

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

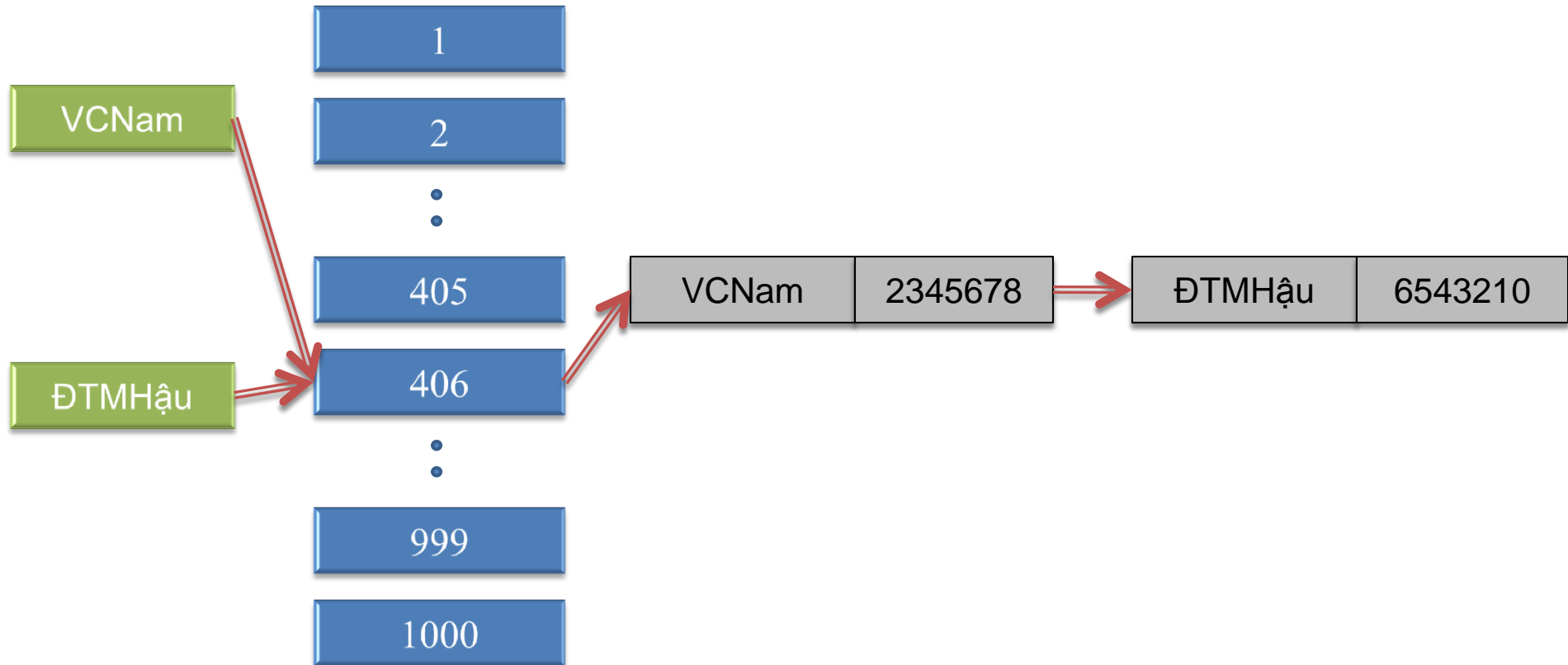
# Phương pháp nối kết

14

- ⊙ 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.

# Giải quyết độ với phương pháp nổi kết

15



# Phương pháp địa chỉ mở

16

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



# Các cách thực hiện

17

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

# Giải quyết độ bằng phương pháp dò tuyến tính

18

- Ý tưởng:  $H(k, i) = (h(k) + i) \bmod M$ , ở đây  $i$  là giá trị thử 0, 1, ..

NTHNhung

1		
2		

⋮

405	NTHNhung	9345678
406		
407		
408		

⋮

999		
1000		

# Giải quyết độ bằng phương pháp dò tuyến tính

19

- Ý tưởng:  $H(k, i) = (h(k) + i) \bmod M$ , ở đây  $i$  là giá trị thử 0, 1, ..

NTHNhung

VCNam

1		
2		

⋮

405	NTHNhung	9345678
406	VCNam	2345678
407		
408		

⋮

999		
1000		

# Giải quyết độ bằng phương pháp dò tuyến tính

20

- Ý tưởng:  $H(k, i) = (h(k) + i) \bmod M$ , ở đây  $i$  là giá trị thử  $0, 1, \dots$

NTHNhung

VCNam

ĐTMHậu

1		
2		

⋮

405	NTHNhung	9345678
406	VCNam	2345678
407	ĐTMHậu	6543210
408		

⋮

999		
1000		

# Giải quyết độ bằng phương pháp dò tuyến tính

21

- Ý tưởng:  $H(k, i) = (h(k) + i) \bmod M$ , ở đây  $i$  là giá trị thử  $0, 1, \dots$

**Tìm?**

VCNam

1		
2		

⋮

405	NTHNhung	9345678
406	VCNam	2345678
407	ĐTMHậu	6543210
408		

⋮

999		
1000		

# Giải quyết độ bằng phương pháp dò tuyến tính

22

- Ý tưởng:  $H(k, i) = (h(k) + i) \bmod M$ , ở đây  $i$  là giá trị thử 0, 1, ..

**Tìm?**

ĐTMHậu

1		
2		

⋮

405	NTHNhung	9345678
406	VCNam	2345678
407	ĐTMHậu	6543210
408		

⋮

999		
1000		

# Phương pháp dò bậc 2 và băm kép

23

- ⊙ *Phương pháp dò bậc 2:*

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

- ⊙ *Phương pháp băm kép:*

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

# Ưu thế của phương pháp địa chỉ mở so với phương pháp nối kết

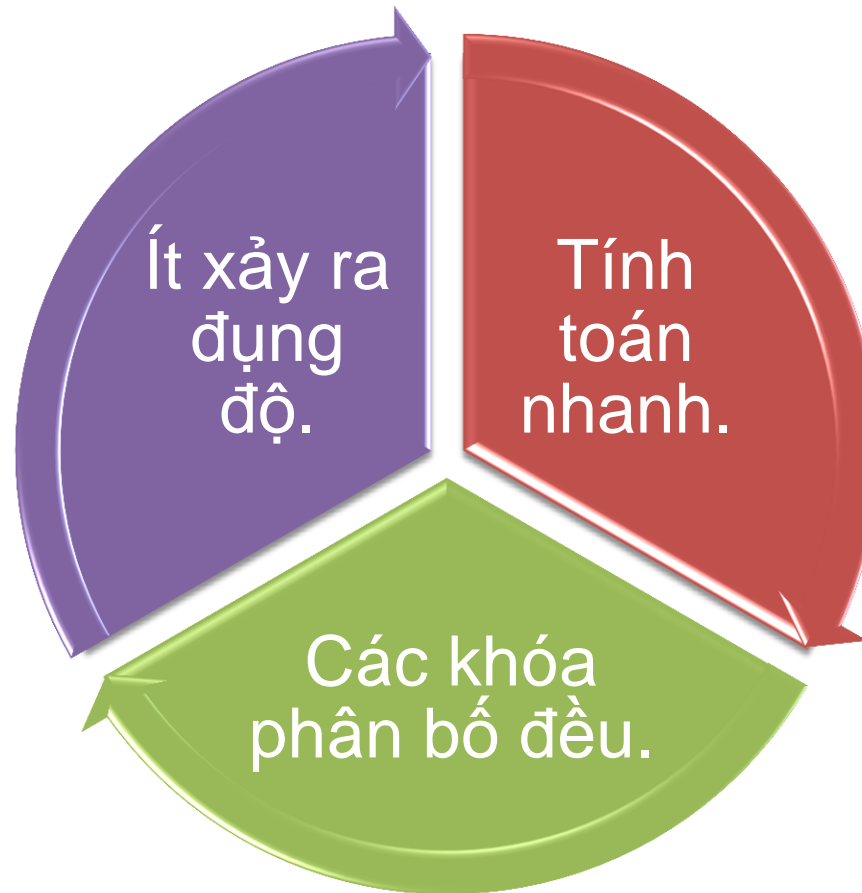
24

- ◉ Đơn giản khi cài đặt.
- ◉ Phương pháp địa chỉ mở 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 tổn bộ nhớ khi mảng thừa (ít phần tử).



# Những yêu cầu đối với hàm băm

25



# Thiết kế hàm băm

26

1. Thiết kế hàm băm cho ứng dụng tra cứu từ điển Anh – Việt.

# Bài tập

27

1. Cho bảng băm có kích thước  $M = 11$ . Hàm băm:  $h(k) = k \bmod M$ . 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  $h_2(k) = (k \bmod 19) + 1$ .

# Bài tập

28

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.

# Hỏi và Đáp