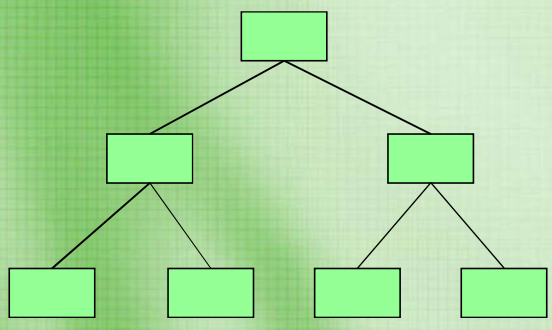
Nội dung (giới thiệu 1-2 tiết)

- Giới thiệu
- Mô tả
- Hàm băm
- Bảng băm kết nối trực tiếp
- Bảng băm kết nối hợp nhất
- Bảng băm dò tìm tuyến tính

Giới thiệu



Tất cả các thao tác phải so sánh khoá!!!

Khắc phục?

Vấn đề cơ bản

- Bài toán: cần lưu trữ các mẫu tin và thực hiện các thao tác
  - Thêm mẫu tin
  - Xoá mẫu tin
  - Tìm mẫu tin theo khóa
- Tìm cách thức thực hiện một cách hiệu quả?

Vấn đề cơ bản

- Unsorted array
  - Sử dụng mảng để lưu mẫu tin, không có thứ tự
  - Thêm: thêm cuối nhanh O(1)
  - Xoá: chậm do tìm rồi xoá O(n)
  - Tìm kiếm: tuần tự chậm O(n)

Vấn đề cơ bản

- Sorted array
  - Sử dụng mảng lưu trữ mẫu tin, có thứ tự
  - Thêm: chèn vào đúng vị trí, chậm O(n)
  - Xoá: phải dời các phần tử phía sau, chậm O(n)
  - Tìm: nhị phân, nhanh O(logn)

Vấn đề cơ bản

- Linked list
  - Lưu trữ mẫu tin trong danh sách liên kết
  - Thêm: nhanh, O(1)
  - Xoá: nhanh khi xoá nút, chậm khi tìm O(n)
  - Tìm kiếm: tìm kiếm tuần tự O(n)

Vấn đề cơ bản

- Cấu trúc dữ liệu phức tạp hơn, nhưng thực thi tốt hơn
  - Tree BST
  - Hash table

Array as table

0012345	An	8.15
0033333	Binh	90
0056789	Danh	5.68
...		
9801010	Phuong	2.0
9802020	Minh	10.0
...		
9903030	Thao	7.3
9908080	Tung	4.9

Vấn đề: lưu trữ 1,000 mẫu tin sinh viên và tìm kiếm theo mã số sinh viên.

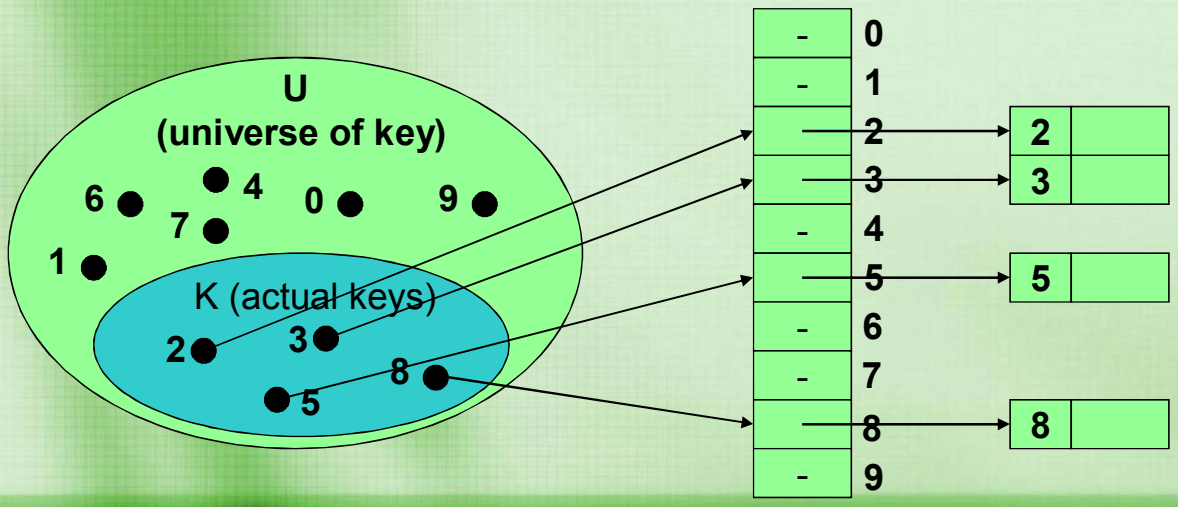
Array as table

0		
:	:	:
12345	An	8.15
:	:	:
33333	Binh	9.0
:	:	:
56789	Danh	5.68
:	:	:
:	:	:
9908080	Tung	4.9
:	:	:
9999999		

Một cách "stupid" là lưu trữ mẫu tin trong mảng cực lớn 0..9999999 Index được sử dụng như là mã số sinh viên. Khi đó mẫu tin sv với ms 0012345 được lưu trữ ở A[12345]!

Array as table

- Dạng bảng băm với địa chỉ trực tiếp
  - Mỗi vị trí tương ứng một khoá trong U
  - Nếu 1 phần tử x có khoá k, thì T[k] chứa con trỏ đến x
  - Ngược lại T[k] = Ø được thể hiện là null



Array as table

- Lưu trữ mẫu tin trong mảng lớn: chỉ mục tương đương khóa
- Thêm: rất nhanh O(1)
- Xóa: rất nhanh O(1)
- Tìm: rất nhanh O(1)
- Nhưng lãng phí rất nhiều bộ nhớ!

Hàm băm "hoàn hảo"

```
int Hash(KeyType  key)
```

Giả sử có hàm **"magic" hash**. Nó ánh xạ mã số của 1000 sinh viên vào các số 0..999, ánh xạ one to one. Không có 2 mã số cùng giá trị ánh xạ.

H('0012345') = 134
H('0033333') = 67
H('0056789') = 764
…
H('9908080') = 3

Bảng băm

Để lưu trữ 1 mẫu tin, gọi Hash(stud_id) và lưu trữ vào vị trí Hash(stud_id) trong mảng.
Để tìm một sinh viên, chỉ cần gọi Hash(stud_id).

0			
	:	:	:
3	9908080	Tung	4.9
	:	:	:
67	0033333	Binh	9.0
	:	:	:
134	0012345	An	8.15
	:	:	:
764	0056789	Danh	5.68
	:	:	:
999	:	:	:

Bảng băm với hàm băm hoàn hảo

- Magic hash
  - Thêm: rất nhanh O(1)
  - Xóa: rất nhanh O(1)
  - Tìm: rất nhanh O(1)
- Thực tế rất khó xây dựng được hàm băm hoàn hảo (khi không gian khóa quá lớn)

Hàm băm

- Hàm băm: biến đổi khóa thành chỉ mục trên bảng băm
  - Khóa có thể là dạng số hay dạng chuỗi
  - Chỉ mục được tính từ 0..M-1, với M là số chỉ mục của bảng băm
  - Hàm băm thường dùng: **key % M**, với M là độ lớn của bảng băm
- Hàm băm tốt phải thoả yêu cầu
  - Giảm thiểu xung đột
  - Phân bố đều trên M địa chỉ khác nhau của bảng băm

Ưu điểm bảng băm

- Dung hòa tốt giữa thời gian truy xuất và dung lượng bộ nhớ
  - Nếu ko giới hạn bộ nhớ: one-to-one, truy xuất tức thì
  - Nếu dung lượng bộ nhớ có giới hạn thì tổ chức khóa cùng địa chỉ
- Bảng băm ứng dụng nhiều trong thực tế, thích hợp tổ chức dữ liệu có kích thước lớn và lưu trữ ngoài

Bảng băm

- Mỗi bảng băm ta cần xác định:
  - Tập khóa k
  - Tập địa chỉ M
  - Hàm băm
- Khi xây dựng mong muốn các khóa ánh xạ vào các địa chỉ khác nhau
- Thực tế thường xảy ra hai khóa cùng địa chỉ $\Rightarrow$ Gọi là **xung đột (collision)**
- Phải có chiến lược giải quyết xung đột!

BB với PP kết nối trực tiếp

- Mỗi địa chỉ bảng băm tương ứng một DSLK
- Các nút bị xung đột nối kết với nhau trên một DSLK
- Danh sách liên kết từ 0 đến M-1
- Khi tìm địa chỉ hàm băm f(key) sẽ xác định địa chỉ i $\in$[0..M-1], ứng với danh sách thứ i chứa phần tử $\Rightarrow$ tiếp tục tìm kiếm trên danh sách liên kết

BB với PP kết nối trực tiếp

- Bảng băm có tập khóa là số nguyên, tập địa chỉ có 10 địa chỉ

BB với PP kết nối trực tiếp

```
#define M 10
typedef struct node
{
        int       Key;
        struct node *    Next;
} NODE;
typedef NODE * NodePtr;
NodePtr          bucket[M];
```

Khóa

Nút kế tiếp (trùng địa chỉ)

10 danh sách liên kết

```
int Hash(int key)
{
        return   key % M;
}
```

Các bảng băm phổ biến khác

- Bảng băm với PP kết nối hợp nhất
- Bảng băm với PP dò tuyến tính
- Sinh viên đọc thêm tài liệu:
  - Nguyễn Hồng Chương, CTDL ứng dụng và cài đặt bằng C, NXB TPHCM, 2005, **p413**
  - **http://en.wikipedia.org/wiki/Hash_table**
  - **http://www.sparknotes.com/cs/searching/hashtables/section1.html**
  - **http://www.cs.auckland.ac.nz/software/AlgAnim/hash_tables.html**

Bài tập

Viết một CT hiện thực từ điển Anh - Việt. Mỗi nút của bảng băm có khai báo các trường sau:

- Trường word là khoá chứa một từ tiếng anh.
- Trường mean là nghĩa tiếng Việt.
- Trường next là con trỏ chỉ nút kế bị xung đột.

Tập khoá là một chuỗi tiếng anh, tập địa chỉ có 26 chữ cái. Chọn hàm băm sau cho khoá bắt đầu bằng ký tự a được băm vào địa chỉ 0, b băm vào địa chỉ 1,…, z băm vào địa chỉ 25.

Chương trình có những chức năng như sau:

1. Nhập vào một từ
2. Xem từ điển theo ký tự đầu.
3. Xem toàn bộ từ điển.
4. Tra từ điển.
5. Xoá một từ, xóa toàn bộ từ điển.