

| | | |
|---|-----------------------------------|--|
| Trường: ĐH CNTP TP.HCM Khoa: Công nghệ thông tin Bộ môn: Công nghệ phần mềm. Môn học: TH Cấu trúc dữ liệu & giải thuật | BÀI 10. BẢNG BĂM | |
|---|-----------------------------------|--|

A. MỤC TIÊU:

- Trình bày được cấu trúc và ý nghĩa của bảng băm.
- Xây dựng được bảng băm lưu trữ phần tử có khóa là số nguyên.
- Giải thích được khái niệm đụng độ và xử lý được bằng phương pháp kết nối trực tiếp.

B. DỤNG CỤ - THIẾT BỊ THÍ NGHIỆM CHO MỘT SV:

| STT | Chủng loại – Quy cách vật tư | Số lượng | Đơn vị | Ghi chú |
|-----|------------------------------|----------|--------|---------|
| 1 | Computer | 1 | 1 | |

C. NỘI DUNG THỰC HÀNH

I. Tóm tắt lý thuyết

1. Khái niệm về bảng băm

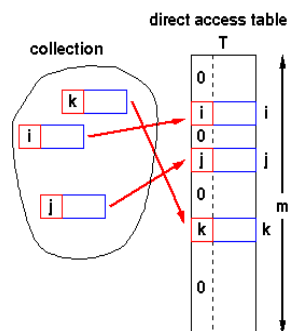
Bảng băm (HashTable) là một cấu trúc mà khi người dùng thực hiện truy xuất một phần tử qua khóa thì nó sẽ được ánh xạ vào thông qua hàm băm (Hash function).

Bảng gồm m phần tử được lưu trữ dưới dạng bảng chỉ mục:

- Phần tử có giá trị khóa k được lưu trữ tương ứng tại vị trí thứ $(k+1)$.
- Tìm kiếm bằng cách tra trong bảng chỉ mục.
- Thời gian tìm kiếm là $O(1)$

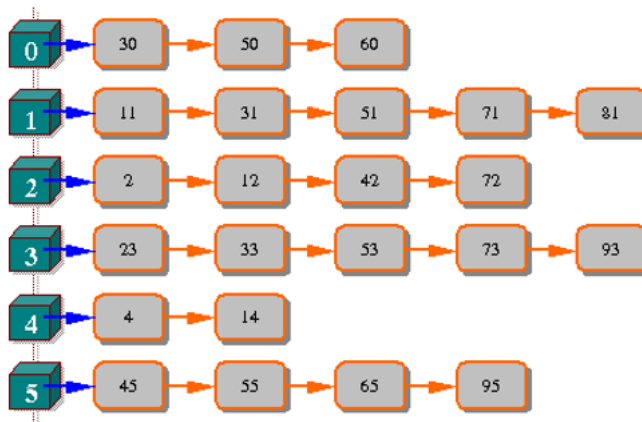
Hàm băm (Hash function) là hàm thực hiện việc ánh xạ khóa k nào đó vào trong bảng băm ($h(k)$). Một hàm băm tốt thỏa mãn các tiêu chí sau:

- Tốc độ tính toán nhanh
- Các khóa được phân bố đều trong bảng
- Ít xảy ra đụng độ



Ví dụ: hàm băm: $h(k) = K \% 100$.

Có nhiều phương pháp để giải quyết đụng độ, trong nội dung của bài học này chúng ta sẽ giải quyết đụng độ bằng **phương pháp kết nối trực tiếp**: Đối với việc sử dụng phương pháp kết nối trực tiếp, các phần tử bị đụng độ sẽ được thêm vào danh sách liên kết tại $h(k)$ trong bảng băm.



2. Cấu trúc bảng băm

Do dùng danh sách để các giá trị trong bảng băm nên cấu trúc của bảng băm kích thước m sẽ gồm:

- Một mảng có m phần tử
- Mỗi phần tử là 1 danh sách liên kết đơn

3. Các thao tác trên cây nhị phân tìm kiếm

a. Khai báo nút

```
//khai báo kích thước bảng băm
#define KEYSIZE 27

//Khai báo cấu trúc node DSLK
struct HashNode
{
    int Key;
    HashNode* Next;
};

//khai báo bảng băm
HashNode* bucket[KEYSIZE];
```

b. Một số thao tác trên bảng băm

```
//Hàm băm
int HashFunction(int Key)
{
    return (Key % KEYSIZE);
}

//Phép toán khởi tạo (InitBucket
void InitBucket()
{
    for (int b = 0; b < KEYSIZE; b++)
        bucket[b] = NULL;
}

//thêm một node vào đầu bucket
void Push(int b, int x) {
    HashNode* p = new HashNode;
    p->Key = x;
    p->Next = bucket[b];
    bucket[b] = p;
}

// Phép toán thêm vào bucket một Node mới sau Node p
void InsertAfter(HashNode* p, int k) {
    if (p == NULL)
        printf("Khong them vao Node moi duoc");
    else {
        HashNode* q = new HashNode;
        q->Key = k;
        q->Next = p->Next;
        p->Next = q;
    }
}

//Phép toán chèn khóa k vào danh sách liên kết :
void Place(int b, int k) {
    HashNode* p, *q;
    q = NULL;
    for (p = bucket[b]; p != NULL && k > p->Key; p = p->Next)
        q = p;
    if (q == NULL)
        Push(b, k);
    else
        InsertAfter(q, k);
}
```

```

//Kiểm tra bucket rỗng(IsEmptyBucket)
int IsEmptyBucket(int b) {
    return(bucket[b] == NULL ? 1 : 0);
}

//Xóa một nút ở đầu bucket :
int Pop(int b) {
    HashNode* p;
    int k;
    if (IsEmptyBucket(b)) {
        printf("Bucket %d bi rong, khong xoa duoc", b);
        return 0;
    }
    p = bucket[b];
    k = p->Key;
    bucket[b] = p->Next;
    delete p;
    return k;
}

//Xóa một nút ngay sau nút p
int DeleteAfter(HashNode* p) {
    HashNode* q;
    int k;
    if (p == NULL || p->Next == NULL) {
        printf("\n Khong xoa Node duoc");
        return 0;
    }
    q = p->Next;
    k = q->Key;
    p->Next = q->Next;
    delete q;
    return k;
}

//Xóa phần tử có khóa k trong bảng băm
void Remove(int k)
{
}

```

```

//Tìm kiếm phần tử chứa khóa k trong bảng băm
HashNode* Search(int k)
{
    ...
}

//Duyệt toàn bộ bảng băm
void Traverse()
{
    ...
}

```

II. Bài tập hướng dẫn mẫu

Hoàn chỉnh các hàm để xây dựng cấu trúc bảng băm chứa khóa là các số nguyên hoàn chỉnh.

III. Bài tập ở lớp

Bài 1. Thông tin của một sinh viên gồm: MSSV, Họ và tên, Năm sinh, Điểm, biết rằng điểm của sinh viên là một số nguyên có giá trị từ 1 đến 100. Hãy xây dựng cấu trúc bảng băm theo phương pháp kết nối trực tiếp để lưu thông tin sinh viên dựa vào điểm số và thực hiện các yêu cầu sau:

- Nhập thông tin sinh viên vào bảng băm từ file
- Xuất thông tin tất cả sinh viên theo nhóm có cùng điểm số
- Liệt kê danh sách sinh viên đạt loại xuất sắc theo thứ tự điểm giảm dần. Biết rằng sinh viên loại xuất sắc là sinh viên có điểm lớn hơn 90.
- Xóa tất cả các sinh viên có điểm bằng x

Bài 2. Xây dựng cấu trúc bảng băm theo phương pháp kết nối trực tiếp để xây dựng một từ điển Anh Việt đơn giản với các chức năng

- Tạo từ điển từ file text chứa dữ liệu gồm n dòng với cấu trúc mỗi dòng như sau:

Từ tiếng Anh: nghĩa tiếng Việt

VD: teacher: xin chào.

- b. Thêm một từ mới vào từ điển
- c. Tra nghĩa của một từ tiếng Anh. Nếu từ đó không có trong từ điển thì xuất ra màn hình "Từ bạn đang tìm không tồn tại!"
- d. Xóa một từ trong từ điển
- e. Xóa toàn bộ từ điển.

Gợi ý: có thể tạo bảng băm có 26 phần tử, lưu các từ có cùng ký tự đầu trong cùng một bucket.

Bài 3. Cho một danh sách các số nguyên có giá trị trong khoảng từ $[1..n]$. Cho một số nguyên k bất kỳ. Sử dụng cấu trúc bảng băm tìm tất cả các cặp số $(a[i], a[j])$ mà có tổng $a[i] + a[j] = k$

IV. Bài tập về nhà

Bài 4. Sử dụng bảng băm để xây dựng từ điển các thuật ngữ chuyên ngành trong tin học

Bài 5. Để đăng nhập vào hệ thống chương trình, mỗi user cần có thông tin về: tên đăng nhập và mật khẩu. Khi đăng nhập người dùng nhập tên đăng nhập và mật khẩu. Hệ thống sẽ báo đăng nhập thành công nếu trùng khớp tên đăng nhập và mật khẩu đã được lưu. Hãy vận dụng bảng băm để kiểm tra tính hợp lệ của một đăng nhập

-- HẾT --