

STRUCT AND ALGORITHMS

3 - 1

HashTable- Bảng Băm

Phần 1: Cơ sở lý thuyết

Xem lại slide bài giảng

Phần 2: Thực hành

Bài 01: Xây dựng cấu trúc bảng băm lưu trữ các số nguyên dương nhập từ bàn phím

```
//Nhập  $N \leq 1000$  số nguyên từ bàn phím.Hãy viết chương trình lần
lượt thêm N giá trị số nguyên trên vào bảng băm có kích thước
m(m>0), sử dụng hàm băm  $H(k) = k \bmod m$ , theo phương pháp dò bậc
hai.
#include <iostream>
using namespace std;
int menu()
{
    int option;
    cout << "-----Menu-----";
    cout << "\n1. Init Hashtable";
    cout << "\n2. Add to Hashtable";
    cout << "\n3. Print Hashtable";
    cout << "\n4. Search Item";
    cout << "\n5. Delete item";
    cout << "\n6. Exit";
    cout << "seleted Option: "; cin >> option;
    return option;
}
//Khởi tạo bảng băm
//typedef float item;
#define max 1001
typedef struct bangbam
{
    int data[max];
    int soPt;
```

```
}BB;
void Init(BB &b)
{
    for (int i = 0; i < max; i++)
    {
        b.data[i] = -1;
        b.soPt = 0;
    }
}
//Hàm băm với value: giá trị cần băm, size: kích thước
int HK(int value, int size)
{
    return (value%size);
}
void Push(BB &b, int value,int size)
{
    int i = HK(value, size);
    while (b.data[i] != -1 && b.soPt<size)//&&, ||
b.data[i]!=value
    {
        i = ((++i)*(i)) % size;
    }
    if (b.soPt < size)
    {
        b.data[i] = value;
        b.soPt++;
    }
    else
    {
        cout << "Hashtable full";
    }
}
void NhapDuLieuChoBangBam(BB &b,int size)
{
    int x;
    do
    {
        cout << "Nhap phan tu thu: " << b.soPt + 1;
        cin >> x;
        if (x != -1)
            Push(b, x,size);
    } while (x!=-1);//Nhap -1 để kết thúc vòng lặp
}
void PrintHashtable(BB b)
{
    for (int i = 0; i < max; i++)
    {
```

```
        if (b.data[i] != -1)
            cout << b.data[i] << " ";
    }
}

int SearchHashtable(BB b, int value, int size)
{
    int i = HK(value, size);
    while (b.data[i] != -1 && b.data[i] != value)
    {
        i = ((++i) * (i)) % size;
    }
    if (b.data[i] == -1)
        return -1; // không tìm thấy
    return i; // thay tại vị trí i
}

int DeleteHashtable(BB &b, int value, int size)
{
    int i = HK(value, size);
    while (b.data[i] != -1 && b.data[i] != value)
    {
        i = ((++i) * (i)) % size;
    }
    if (b.data[i] == -1)
    {
        return -1; // không tìm thấy
    }
    else // tìm thấy
    {
        b.data[i] = -1;
        b.soPt--;
        return i;
    }
}

void main()
{
    int option = 0;
    BB b;
    do
    {
        option = menu();
        switch (option)
        {
            case 1:
                Init(b);
                break;
        }
    }
}
```

```
        case 2:
            NhapDuLieuChoBangBam(b, max);
            break;
        case 3:
            PrintHashtable(b);
            break;
        case 4:
        {
            int value;
            cout << "Nhap gia tri can tim: "; cin >> value;
            int l = SearchHashtable(b, value, max);
            if (l == -1)
            {
                cout << "Khong tim thay " << value << " trong
Hashtable";
            }
            else
            {
                cout << "Tim thay " << value << " tai vi tri "
<< l;
            }
            break;
        }
        case 5:
        {
            int value1;
            cout << "Nhap gia tri can tim: "; cin >> value1;
            int k = DeleteHashtable(b, value1, max);
            if (k == -1)
            {
                cout << "Khong tim thay " << value1 << " trong
Hashtable";
            }
            else
            {
                cout << "Tim thay " << value1 << " tai vi tri
" << k;
            }
            break;
        }
    }
} while (option != 6);
}
```

Bài 2: Sử dụng bảng băm với kiểu dữ liệu tự xây dựng

```
#include <iostream>
using namespace std;
int menu()
{
    int option;
    cout << "-----Menu-----";
    cout << "\n1. Init Hashtable";
    cout << "\n3. Push into Hashtable";
    cout << "\n4. Prints Hashtable";
    cout << "\n5. Get item top Hashtable";
    cout << "\n6. Exit";
    cout << "seleted Option: "; cin >> option;
    return option;
}
//tổ chức dữ liệu để lưu trữ thông tin mặt hàng cần:
typedef struct mathang
{
    int MaMH ;
    char TenMH[30];
    char DVT[20];
    float DoiGia;
    int SoLuong;
}MATHANG;
MATHANG Nhap_01_MatHang(int i)
{
    MATHANG _matHang;
    cout << "Nhap Thong tin mat hang thu: " <<i<< endl;
    cout << "MaMH: "; cin >> _matHang.MaMH;
    cout << "Ten MH: "; fflush(stdin);
    cin.getline(_matHang.TenMH,30);
    cout << "DVT: "; cin>>_matHang.DVT;
    cout << "Don gia: "; cin >> _matHang.DoiGia;
    cout << "So luong: "; cin >> _matHang.SoLuong;
    return _matHang;
}
void XuatThongTin_01_MatHang(MATHANG mathang)
{
    cout << "MaMH: " << mathang.MaMH<<"\t";
    cout << "Ten MH : " << mathang.TenMH << "\t";
    cout << "DVT : " << mathang.DVT << "\t";
    cout << "So luong : " << mathang.SoLuong << "\t";
    cout << "Don gia: " << mathang.DoiGia << endl;
}
//Tạo cấu trúc bảng băm để lưu trữ thông tin mặt hàng.
//value là giá trị cần lưu trữ
//Size: kích thước bảng băm;
#define max 101
```

```
int HK1(int value, int size)
{
    return (value%size);
}
int HK2(int value, int size)
{
    return size - 2 - (value % (size - 2));
}
typedef struct bangbam
{
    MATHANG data[max];
    int soPT;
}BB;
void Init(BB &b,int size)
{
    for (int i = 0; i < size; i++)
    {
        b.data[i].MaMH = -1;
    }
    b.soPT = 0;
}
//Thêm 1 phần tử vào bảng băm
void Push(BB &b, MATHANG value,int size)
{
    int i = HK1(value.MaMH,size);
    int j = HK2(value.MaMH, size);
    while (b.data[i].MaMH!=-1&&b.soPT<size)
    {
        i = (i + j) % size;
    }
    if (b.soPT < size)
    {
        b.data[i] = value;
        b.soPT++;
    }else
    {
        cout << "bang bam day:";
    }
}
void main()
{
    int option = 0;
    do
    {
        option = menu();
        switch (option)
```

```
    {  
        case 1:  
  
            break;  
        case 2:  
  
            break;  
    }  
} while (option != 6);  
}
```

Một số bài tập liên quan đến bảng Băm

1. Một cửa hàng quản lý điện thoại, thông tin một điện thoại gồm:

Mã điện thoại: số nguyên dương

Tên điện thoại: chuỗi 20 kí tự

Đơn vị tính: chuỗi 10 kí tự

Đơn giá: số thực

Số lượng: số nguyên dương.

2. Hãy định nghĩa bảng băm có kích thước M là số nguyên tố nhỏ nhất và lớn hơn N để lưu trữ danh sách gồm $N \leq 100$ điện thoại với khóa là mã điện thoại bằng phương pháp băm kép. Biết rằng hàm băm thứ 2 được tính theo công thức sau:

$$h_2(k) = M - 2 - (k \% (M - 2))$$

Biết rằng k là mã điện thoại. Lập trình cài đặt các chức năng sau trên bảng băm:

Nhập danh sách điện thoại để lưu trữ.

Yêu cầu người dùng nhập vào mã điện thoại cần xóa và hãy xóa nó ra khỏi bảng băm.

Nhập $N \leq 1000$ số nguyên từ bàn phím. Hãy định nghĩa bảng băm có kích thước M là số nguyên tố nhỏ nhất và lớn hơn N để lưu trữ N số nguyên bằng phương pháp dò bình phương. Lập trình cài đặt các chức năng sau trên bảng băm:

a) Lưu trữ N số nguyên trong bảng băm.

b) Liệt kê các số nguyên tố và chỉ mục lưu trữ của các số nguyên tố trong bảng băm.

Ví dụ:

Ở chỉ mục thứ 5 lưu trữ số nguyên tố 2.

Ở chỉ mục thứ 7 lưu trữ số nguyên tố 3.

c) Tìm và xoá giá trị x trong bảng băm. (với x nhập từ bàn phím)

3. Một hội đồng tuyển sinh quản lý các thí sinh dự thi theo các thông tin sau:

- SoBD: chuỗi gồm 5 kí tự số
- Họ tên: chuỗi gồm tối đa 30 kí tự
- Khu vực ưu tiên: chuỗi "KV1", "KV2" hoặc "KV3"
- Điểm thi môn Toán, Lý, Hóa: số thực

Hãy định nghĩa bảng băm có kích thước M là số nguyên tố nhỏ nhất và lớn hơn N để lưu trữ danh sách gồm $N \leq 1000$ thí sinh với khóa là số báo danh bằng phương pháp dò tuyến tính. Lập trình cài đặt các nghiệp vụ sau trên bảng băm:

- Nhập đầy đủ thông tin cho N thí sinh gồm: số báo danh, họ tên, khu vực ưu tiên, điểm môn Toán, Lý, Hóa và lưu trữ vào bảng băm.
- Xuất đầy đủ thông tin của thí sinh với chuỗi số báo danh được cho trước

4. Một công ty quản lý nhân viên theo các thông tin sau:

- Mã nhân viên: chuỗi 6 kí tự số
- Họ tên: chuỗi tối đa 30 kí tự
- Hệ số lương: số thực dương

Hãy định nghĩa bảng băm có kích thước M là số nguyên tố nhỏ nhất và lớn hơn N để lưu trữ danh sách gồm $N \leq 1000$ nhân viên với khóa là mã nhân viên bằng phương pháp băm kép. Biết rằng hàm băm thứ 2 được tính theo công thức sau:

$$h_2(k) = M - 2 - (k \% (M - 2))$$

Biết rằng k là mã nhân viên. Lập trình cài đặt các nghiệp vụ sau trên bảng băm:

Nhập đầy đủ thông tin cho N nhân viên gồm: mã nhân viên, họ tên, hệ số lương và lưu trữ vào bảng băm.

Hủy nhân viên có mã nhân viên được cho trước ra khỏi bảng băm.

5. Một công ty quản lý nhân viên theo các thông tin sau:

- Mã nhân viên: chuỗi 6 kí tự số
- Họ tên: chuỗi tối đa 30 kí tự

- Ngày công: số nguyên dương
- Hệ số lương: số thực dương

Hãy định nghĩa bảng băm có kích thước M là số nguyên tố nhỏ nhất và lớn hơn N để lưu trữ danh sách gồm $N \leq 1000$ nhân viên với khóa là mã nhân viên bằng phương pháp dò tuyến tính. Lập trình cài đặt các nghiệp vụ sau trên bảng băm:

- a) Nhập đầy đủ thông tin cho N nhân viên gồm: mã nhân viên, họ tên, ngày công, hệ số lương và lưu trữ vào bảng băm.
- b) Tìm và xuất đầy đủ thông tin của nhân viên có mã được cho trước kèm theo lương thực lĩnh. Biết rằng lương thực lĩnh = ngày công * 200000 * hệ số lương.