PHƯƠNG PHÁP ĐỊA CHỈ MỞ

❖ PHƯƠNG PHÁP ĐỊA CHỈ MỞ (Open Addressing)

Phương pháp này giải quyết đụng độ bằng thăm dò (probe) từng bước những địa chỉ khác còn trống trên bảng băm kích thước m nhờ một hàm băm thăm dò.

Hàm băm thăm dò f có dạng như sau:

$$f(k, i) = (h(k) + g(k, i)) \% m$$

- h(k) là một hàm băm.
- g(k, i) là một hàm số theo i, k
- i là lần thăm dò

PHƯƠNG PHÁP ĐỊA CHỈ MỞ

❖ PHƯƠNG PHÁP ĐỊA CHỈ MỞ (Open Addressing)

Ví dụ 9: Cho bảng băm dạng đóng kích thước 7, hàm băm thăm dò f(k, i) = ((k % m) + i) % m, Các khóa được đưa vào lần lượt là 1,2,8,15.

PHƯƠNG PHÁP ĐỊA CHỈ MỞ

❖PHƯƠNG PHÁP ĐỊA CHỈ MỞ (Open Addressing)

1, lần thăm dò 0
f(1, 0) = 1

0	
1	1
2	
3	
4	
5	
6	

2, lần thăm dò 0
f(2, 0) = 2

0	
1	1
2	2
3	
4	
5	
6	

8, lần thăm dò 0
f(8, 0) = 1

0	
1	1
2	2
3	
4	
5	
6	

PHƯƠNG PHÁP ĐỊA CHỈ MỞ

❖PHƯƠNG PHÁP ĐỊA CHỈ MỞ (Open Addressing)

8, lần thăm dò 1
f(8, 1) = 2

0	
1	1
2	2
3	
4	
5	
6	

8, lần thăm dò 2
f(8, 2) = 3

0	
1	1
2	2
3	8
4	
5	
6	

15, lần thăm dò 0
f(15, 0) = 1

0	
1	1
2	2
3	8
4	
5	
6	

PHƯƠNG PHÁP ĐỊA CHỈ MỞ

❖PHƯƠNG PHÁP ĐỊA CHỈ MỞ (Open Addressing)

15, lần thăm dò 1
f(15, 1) = 2

0	
1	1
2	2
3	8
4	
5	
6	

15, lần thăm dò 2
f(15, 2) = 3

0	
1	1
2	2
3	8
4	
5	
6	

15, lần thăm dò 3
f(15, 2) = 4

0	
1	1
2	2
3	8
4	15
5	
6	

PHƯƠNG PHÁP ĐỊA CHỈ MỞ

❖CÁC PHƯƠNG PHÁP THĂM DÒ

- Phương pháp thăm dò tuyến tính: nếu hàm g(k, i) là một hàm tuyến tính theo i có dạng

$$g(k, i) = a*i + b.$$

  Có thể chọn g(k, i) = i.

- Phương pháp thăm dò bậc 2 (toàn phương): nếu hàm g(k, i) là một hàm bậc 2 theo i có dạng

$$g(k, i) = a * i^2 + b * i + c.$$

  Có thể chọn $g(k, i) = i^2$

PHƯƠNG PHÁP ĐỊA CHỈ MỞ

❖CÁC PHƯƠNG PHÁP THĂM DÒ

- Phương pháp băm kép: nếu g(k, i) là một hàm số có dạng

$$g(k, i) = i * h_1(k).$$

$h_1(k)$ được gọi là hàm băm phụ và cần được xây dựng theo kích thước bảng băm m như sau:

  - Trường hợp $m = 2^p$, $h_1(k)$ cần trả về giá trị lẻ.
  - Trường hợp m là số nguyên tố, $h_1(k)$ cần có miền giá trị là (0, m)

PHƯƠNG PHÁP ĐỊA CHỈ MỞ

❖CÁC PHƯƠNG PHÁP THĂM DÒ

Ví dụ 9: cho bảng băm đóng kích thước 7 dùng phương pháp băm kép, hàm băm phụ cần có dạng:

$$h_1(k) = 1 + (k \,\%\, 6)$$

Ví dụ 10: cho bảng băm đóng kích thước 8 dùng phương pháp băm kép, hàm băm phụ cần có dạng:

$$h_1(k) = (k\%8) + ((k+1)\%2)$$

PHƯƠNG PHÁP ĐỊA CHỈ MỞ

❖TỔ CHỨC DỮ LIỆU

```
#define DEL -1
#define EMPTY 0
// giả sử khóa có giá trị là số nguyên dương.
struct CHashtable {
    int m, n;
    int * buckets; // giả sử bảng băm chỉ lưu khóa
};
```

PHƯƠNG PHÁP ĐỊA CHỈ MỞ

❖CÁC THAO TÁC

- Tạo bảng băm đóng

```
void CreateCHashtable(CHashtable &ht, int m) {
    ht.buckets = new int[m];
    if (ht.buckets == NULL) m = 0;
    else {
    for (int i = 0; i < m; i++) ht.buckets[i] = EMPTY;
        ht.m = m;
    }
    ht.n = 0;
}
```

PHƯƠNG PHÁP ĐỊA CHỈ MỞ

❖CÁC THAO TÁC

- Đưa phần tử có khóa x vào bảng băm

```
int g(int key, int m, int i);     // hàm tuyến tính, bậc 2 hay hàm
                                  // băm phụ
int h(int key, int m);            // hàm băm dạng chia hay nhân
int CHashCode(int key, int m, int i) {
    return (h(key, m) + g(key, m, i)) % m;
}
```

PHƯƠNG PHÁP ĐỊA CHỈ MỞ

```
int CPutKey(CHashtable &ht, int key) {
    if (ht.m == 0 || ht.n == ht.m) return 0;
    int i = 0, k;
    do {
        k = CHashCode(key, ht.m, i); i++;
    } while (((ht.buckets[k] != EMPTY) ||
                (ht.buckets[k] != DEL)) && (i <= ht.m));
    if (i >= ht.m) return 0;
    ht.buckets[k] = key; ht.n++;
    return 1;
}
```

PHƯƠNG PHÁP ĐỊA CHỈ MỞ

❖CÁC THAO TÁC

- Lấy địa chỉ của phần tử có khóa x

```
int CGetKey(CHashtable ht, int key) {
    if (ht.m == 0) return -1;
    int i = 0, k;
    do {
        k = CHashCode(key, ht.m, i); i++;
    } while ((i <= ht.m) && (ht.buckets[k] != key) &&
                            (ht.buckets[k] != EMPTY));
    if (ht.buckets[k] == key) return k; else return -1;
}
```

PHƯƠNG PHÁP ĐỊA CHỈ MỞ

❖CÁC THAO TÁC

- Lấy phần tử tại bucket i

```
int CGetBucket(CHashtable ht, int i) {
    if (ht.m == 0) return EMPTY;
    return ht.buckets[i];
}
int isFull(CHashtable ht) {
    return ht.n == ht.m - 1;
}
```

PHƯƠNG PHÁP ĐỊA CHỈ MỞ

❖CÁC THAO TÁC

- Xóa phần tử có khóa x

```
void CRemoveKey(CHashtable &ht, int key) {
   if (ht.m == 0) return;
   int i = 0, k;
   do {
        k = CHashCode(key, ht.m, i); i++;
   } while ((i <= ht.m) && (ht.buckets[k] != key) &&
            (ht.buckets[k] != EMPTY));
   if ((i >= ht.m) || (ht.buckets[k] == EMPTY)) return;
   ht.buckets[k] = DEL; ht.n--; }
```

PHƯƠNG PHÁP ĐỊA CHỈ MỞ

❖CÁC THAO TÁC

- Hủy toàn bộ bảng băm

```
void RemoveCHashtable(CHashtable &ht) {
    if (ht.m == 0) return;
    delete[] ht.buckets;
    ht.buckets = NULL; m = 0;
}
```

PHƯƠNG PHÁP ĐỊA CHỈ MỞ

❖ĐẶC ĐIỂM

- Số phần tử cố định
- Mỗi khóa ứng với một địa chỉ
- Thời gian truy xuất thấp.
- Xóa một phần tử không thu hồi được vùng nhớ của nó.
- Luôn chừa 1 phần tử trống trong bảng băm, nghĩa là nếu bảng băm có kích thước m thì sẽ đầy khi số phần tử trong bảng băm là m-1.

LƯU Ý

Để tính quá trình tính toán địa chỉ trên bảng băm của một khóa k vào lần thăm dò thứ i được hiệu quả, cần cài đặt các thao tác thêm, tìm kiếm và xóa cụ thể cho từng phương pháp thăm dò. (xem Giáo trình)