

CANTHO UNIVERSITY

**Cấu trúc dữ liệu - CT177**

**Chương 5**

**Tự điển**

B môn Công Nghệ Phần Mềm

Khoa CNTT&TT - Đại học Cần Thơ

www.ctu.edu.vn

CANTHO UNIVERSITY

**NỘI DUNG**

- Khái niệm
- Cài đặt liên kết
- Cài đặt liên kết băm

www.ctu.edu.vn

CANTHO UNIVERSITY

**TỰ ĐIỂN**

- **T** **i** **n** là một kỹ thuật lưu trữ thông tin để thực hiện các phép toán thêm (INSERT), bớt (DELETE) và tìm kiếm (MEMBER) có hiệu quả.
- **T** **i** **n** có thể cài đặt bằng:
  - Danh sách liên kết.
  - Băm.

www.ctu.edu.vn

CANTHO UNIVERSITY

**CÀI ĐẶT TỰ ĐIỂN BẰNG MẢNG –**  
**Danh sách đặc**

• **Khai báo**


```
#define MaxLength ...
typedef ... ElementType;
typedef int Position;
typedef struct
{
    ElementType Data[MaxLength];
    Position Last;
} SET;
```

Diagram illustrating the array-based implementation of a dictionary:

0	Phần tử đầu tiên	1
1	Phần tử thứ hai	2
2	...	3
...	...	...
Last-1	Phần tử cuối cùng	Last

Labels: Maxlength-1, Chỉ số, Mảng, Data, Vị trí, MaxLength.

www.ctu.edu.vn



## CÀI ĐẶT TỰ ĐIỂN BẰNG MẢNG

---

- Khởi tạo

```

void makenullSet(SET *L){
    (*L).Last=0;
    L->Last=0;
}

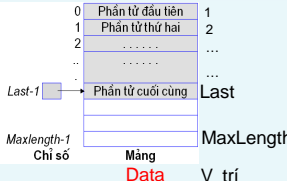
```

- Kiểm tra 1 phần tử có trong tập hay không

```

int isMember(ElementType X, SET L){
    Position P=1, Found=0;
    while ((P <= L.Last) && (Found == 0))
        if ((L.Data[P-1]) == X) Found = 1;
        else P++;
    return Found;
}

```



Mảng Data V trí

[www.ctu.edu.vn](http://www.ctu.edu.vn)



## CÀI ĐẶT TỰ ĐIỂN BẰNG MẢNG

---

- Thêm 1 )

[www.ctu.edu.vn](http://www.ctu.edu.vn)

## CÀI ĐẶT TỰ ĐIỂN BẰNG BẢNG BĂM

- Hàm băm** là mánh xít t p đ li u A n các s nguyên 0..B-1. Hàm băm c s đ ng tìm giá tr b m.
- Các ph ng pháp xác nh hàm băm
  - Ph ng pháp chia
 

x	H(x)= x mod B với B=10
34	4
19	9
  - Ph ng pháp nhân
 

x	x <sup>2</sup>	H(x) gồm 3 số ở giữa
5402	29181604	181 ho c 816
0367	00134689	134 ho c 346
  - Ph ng pháp tách
    - Tách khóa. VD 17046329 có giá tr b m (329+046+017)%1000 = 392
    - G p khóa. VD 17046329 có giá tr b m (923+046+710)%1000 = 679

www.ctu.edu.vn

## BĂM ĐÓNG

- B ng b m óng** l u gi các ph n t c a t i n ngay trong m ng.
- Bucket th i ch a ph n t có giá tr b m là i.
- N u có nhi u ph n t có cùng giá tr b m, **chiến lược băm lại** (rehash strategy) c s đ ng gi i quy t s ng .

Ch s

0	20
1	
2	
3	
4	34
5	
6	26
...	
B-1 (9)	19

Hàm băm:  $H(x)=x\%B$

Ví d hàm băm l i tụy n tính  
 $H_i(x)=(H(x)+i) \bmod B$

www.ctu.edu.vn

## BĂM ĐÓNG

- Khai báo**

```
#define B 100
#define Deleted -1000
//Gia dinh gia tri cho o da bi xoa
#define Empty 1000
//Gia dinh gia tri cho o chua su dung
typedef int ElementType;
typedef ElementType Dictionary[B];
```

www.ctu.edu.vn

## BĂM ĐÓNG

- T ot i n r ng**

```
void makenullDic(Dictionary D)
{
    for (int i=0 ;i<B; i++)
        D[i]=Empty;
}
```
- Hàm băm**

```
int H(ElementType X)
{
    return X%B;
}
```

0	E
1	E
2	E
3	E
4	E
5	E
6	E
...	E
B-1 (9)	E

E: Empty

www.ctu.edu.vn

**BẦM ĐÓNG**

• Ví dụ: Kiểm tra xem giá trị 29, 39 có trong bảng băm với B=10

Ch s	0	20
	1	29
	2	E
	3	E
	4	34
	5	E
	6	26
	7	E
	8	E
B-1 (9)		19

Ví dụ hàm băm l i tùy n tính  
 $H_i(x) = (H(x) + i) \bmod B$   
 $i=0 \rightarrow H(29) = 9$   
 $i=1 \rightarrow H(29) = 0$   
 $i=2 \rightarrow H(29) = 1$

$H(x) = x \% B$

www.ctu.edu.vn

**BẦM ĐÓNG**

• Kiểm tra tính thành phần trong bảng băm

```

int isMember(ElementType X, Dictionary D)
{
    int i=0, init=H(X);
    while( (i<B) && (D[(i+init)%B] != Empty)
        && (D[(i+init)%B] != X) )
        i++;
    return (D[(i+init)%B] == X);
}

```

www.ctu.edu.vn

**BẦM ĐÓNG**

• Ví dụ: Ta cần lưu trữ các số nguyên 34, 20, 26 và 19 vào trong bảng băm có số bucket B = 10 và sử dụng hàm băm  $h(x) = x \% 10$

Ch s	0	20
	1	E
	2	E
	3	E
	4	34
	5	E
	6	26
	7	E
	8	E
B-1 (9)		19

$H(x) = x \% B$

www.ctu.edu.vn

**BẦM ĐÓNG**

• Ví dụ: Thêm giá trị 29 vào bảng băm có B=10 và sử dụng hàm băm l i  $H_i(x) = (H(x) + i) \bmod B$  giải quyết trường hợp va chạm.

Ch s	0	20
	1	29
	2	E
	3	E
	4	34
	5	E
	6	26
	...	E
		E
B-1		19

$H(x) = x \% B$

Ví dụ hàm băm l i tùy n tính  
 $H_i(x) = (H(x) + i) \bmod B$   
 $i=0 \rightarrow H(29) = 9$   
 $i=1 \rightarrow H(29) = 0$   
 $i=2 \rightarrow H(29) = 1$

www.ctu.edu.vn

**BẮM ĐÓNG**

• Ví dụ : Thêm vào giá trị 30

0	20
1	29
2	12
3	30
4	34
5	E
6	26
...	E
B-1	E
	19

$i=0 \rightarrow H(30) = 0$   
 $i=1 \rightarrow H(30) = 1$   
 $i=2 \rightarrow H(30)=2$   
 $i=3 \rightarrow H(30)=3$

www.ctu.edu.vn

**BẮM ĐÓNG**

• Thêm phần tử vào bảng băm

```
void insertDic(ElementType X, Dictionary D)
{
    int i=0,init;
    if (fullDic(D))
        printf("Bang bam day");
    else if (isMember(X,D)==0){
        init=H(X);
        while((i<B)&&(D[(i+init)%B]!=Empty)&&
            (D[(i+init)%B]!=Deleted))
            i++;
        D[(i+init)%B]=X;
    }
    else
        printf("\nPhan tu da ton tai");
}
```

www.ctu.edu.vn

**BẮM ĐÓNG**

• Bài tập: Giả sử bảng băm có 7 bucket, hàm băm là  $h(x) = x \bmod 7$ . Hãy vẽ hình biểu diễn bảng băm khi ta lần lượt đưa vào bảng băm các khóa 1, 8, 27, 64, 125, 216, 343 trong trường hợp dùng bảng băm đóng với chỉ số băm gì quy tắc là phép thay thế n tính?

www.ctu.edu.vn

**BẮM ĐÓNG**

• Ví dụ : Xóa giá trị 30

0	20
1	29
2	12
3	Deleted
4	34
5	E
6	26
...	E
B-1	E
	19

$i=0 \rightarrow H(30) = 0$   
 $i=1 \rightarrow H(30) = 1$   
 $i=2 \rightarrow H(30)=2$   
 $i=3 \rightarrow H(30)=3$

www.ctu.edu.vn

**BĂM ĐÓNG**

- Xóa phần tử ra khỏi bảng băm

```
void deleteDic(ElementType X, Dictionary D)
{
    if (emptyDic(D))
        printf("\nBảng băm rỗng!");
    else
    {
        int i=0, init=H(X);
        while ((i<B)&&(D[(i+init)%B]!=X)&&
            (D[(i+init)%B]!=Empty))
            i++;
        if (D[(i+init)%B]==X)
            D[(i+init)%B]=Deleted;
    }
}
```

www.ctu.edu.vn

**BĂM MỞ**

- Khai báo

```
#define B ...
typedef ... ElementType;
struct Node
{
    ElementType Data;
    struct Node* Next;
};
typedef struct Node* Position;
typedef Position Dictionary[B];
```

www.ctu.edu.vn

**BĂM MỞ**

- Tạo mảng

```
void makenullDic(Dictionary *D)
{
    for(int i=0; i<B; i++)
        (*D)[i]=NULL;
}
```

0	*
1	*
2	*
...	*
B-1	*


www.ctu.edu.vn

**BĂM MỞ**

- Kiểm tra sự tồn tại của phần tử trong bảng băm

```
int isMember(ElementType X, Dictionary D)
{
    Position P;
    int Found=0;
    P=D[H(X)]; //Tìm ô mục H(X)
    //Duyệt trên ds thu H(X)
    while((P!=NULL) && (!Found))
        if (P->Data==X) Found=1;
        else P=P->Next;
    return Found;
}
```

www.ctu.edu.vn




## BẮM MỜ

- Thêm phần tử vào từ điển

```
void insertDic(ElementType X, Dictionary *D)
{
    int Bucket;
    Position P;
    if (!isMember(X, *D)) {
        Bucket = H(X);
        P = (*D)[Bucket];
        //Cap phat o nho moi cho *D[Bucket]
        (*D)[Bucket] = (struct Node*) malloc(sizeof(struct Node));
        (*D)[Bucket] -> Data = X;
        (*D)[Bucket] -> Next = P;
    }
}
```


www.ctu.edu.vn



## BẮM MỜ

- Bài tập: Giả sử bảng băm có 7 bucket, hàm băm là  $h(x) = x \bmod 7$ . Hãy vẽ hình biểu diễn bảng băm khi ta lần lượt đưa vào bảng băm rặng các khoá 1, 8, 27, 64, 125, 216, 343 trong trình hình p dùng bảng băm?

www.ctu.edu.vn



## XÓA PHẦN TỬ RA KHỎI BẢNG BẮM MỜ

```
void deleteDic(ElementType X, Dictionary *D) {
    int Bucket, Done;
    Position P, Q;
    Bucket = H(X);
    if ((*D)[Bucket] != NULL) // danh sách tồn tại
    {
        if ((*D)[Bucket] -> Data == X) // X đầu danh sách
        {
            Q = (*D)[Bucket];
            (*D)[Bucket] = (*D)[Bucket] -> Next;
            free(Q);
        }
        else // Tìm X
        {
            Done = 0;
            P = (*D)[Bucket];
            while ((P -> Next != NULL) && (!Done))
            {
                if (P -> Next -> Data == X) Done = 1; else P = P -> Next;
            }
            if (Done) // Nếu tìm thấy
            {
                Q = P -> Next; // Xóa P -> Next
                P -> Next = Q -> Next;
                free(Q);
            }
        }
    }
}
```

www.ctu.edu.vn




www.ctu.edu.vn