LAB. 09
HÀM BĂM - BẢNG BĂM

MỤC TIÊU

Hoàn tất bài thực hành này, sinh viên có thể:

- Hiểu và sử dụng bảng băm.
- Xây dưng được hàm băm.
- Hiểu và vận dụng các thuật toán xử lý đụng độ.

THỜI GIAN THỰC HÀNH

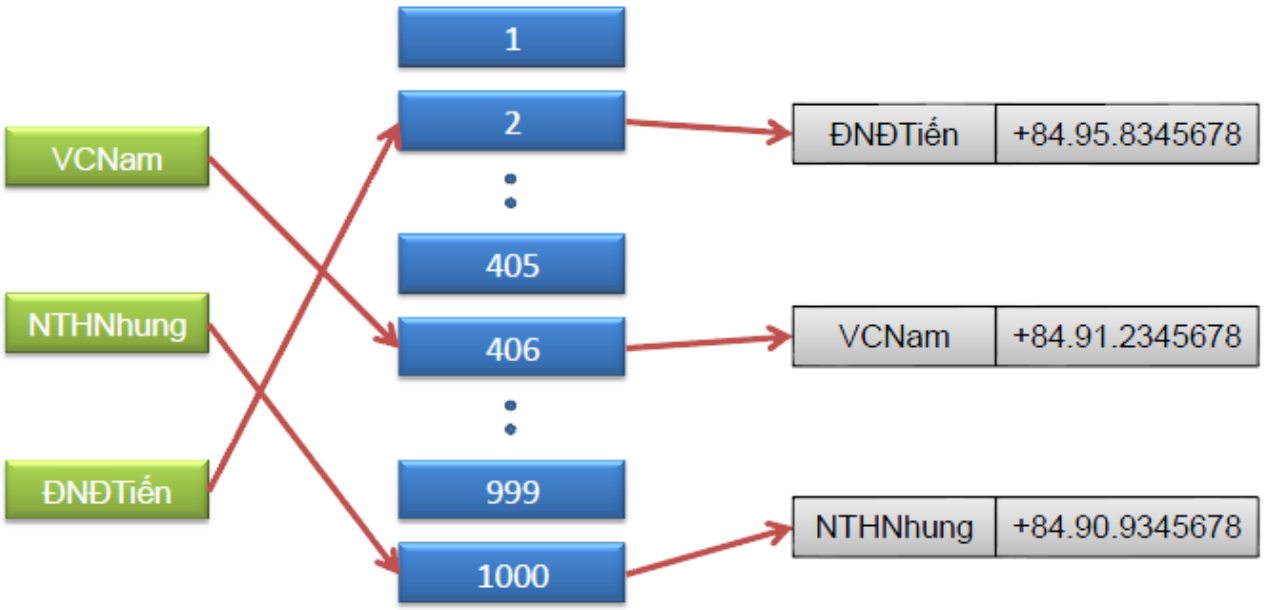
Từ 120phút-240phút

1. KHÁI NIỆM VỀ HASH

Vấn đề: Cho trước một tập S gồm các phần tử được đặc trưng bởi giá trị khoá. Trên giá trị các khoá này có quan hệ thứ tự. Tổ chức S như thế nào để tìm kiếm 1 phần tử có khoá k cho trước có độ phức tạp ít nhất trong giới hạn bộ nhớ cho phép?

Ý tưởng: Biến đổi khoá k thành một số (bằng hàm hash) và sử dụng số này như là địa chỉ để tìm kiếm trên bảng dữ liệu.

Ví dụ:



2. ĐỊNH NGHĨA HÀM BĂM

Định nghĩa: là hàm biến đổi khoá k của phần tử thành địa chỉ trong bảng băm.

Ví dụ: **H(k) = k mod M.**

Tổng quát về phép biến đổi khoá: Là 1 ánh xạ thích hợp từ tập các khoá U vào tập các địa chỉ A.

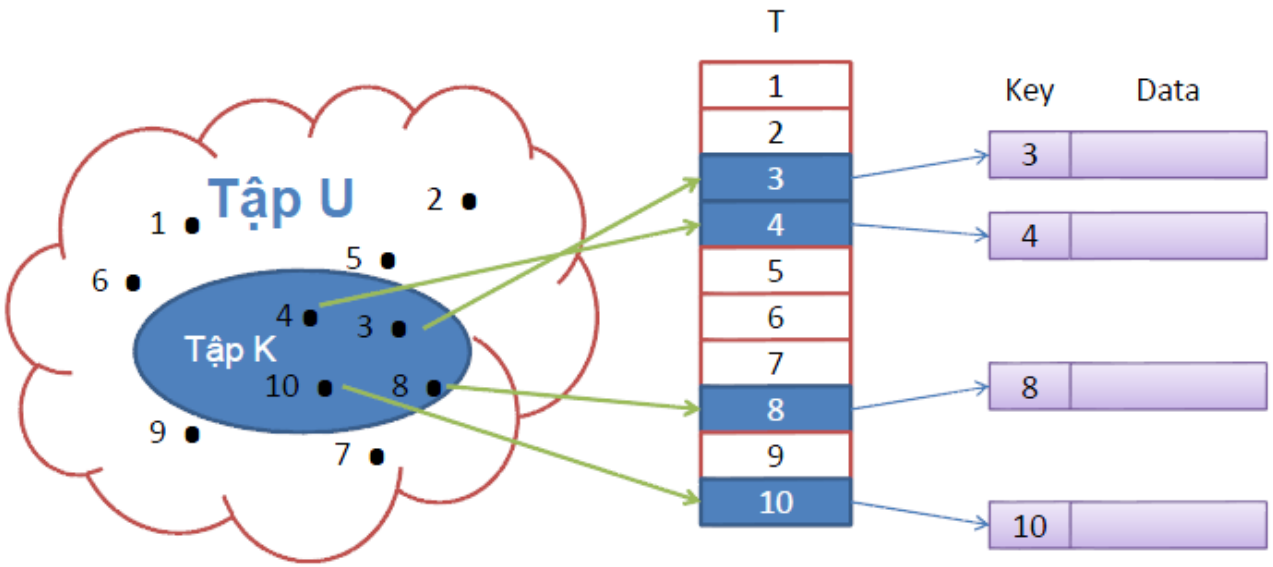
H: U → A

K → a = h(k)

Tập các giá trị khoá (U) có thể lớn hơn rất nhiều so với số khoá thực tế (K).

Ví dụ: Quản lý danh sách của 1000 sinh viên, MSSV gồm 7 chữ số.

Như vậy sẽ có U = 10^7 khoá so với K = 1000.



Bảng băm rất thích hợp cho các bài toán "từ điển (dictionary)". Các bài toán dạng này chỉ chủ yếu sử dụng hai thao tác: chèn (insert) và tìm kiếm (search).

3. XÂY DỰNG HÀM BĂM

3.1 Hàm băm cơ bản

Bài toán: Cho một mảng các số nguyên U có M phần tử. U chính là bảng băm. Có k khoá cũng là các số nguyên, mỗi khoá được lưu vào bảng băm sử dụng công thức tính modulo *(H(k) = k mod M)*. Sau đó nhập một giá trị khoá, in ra giá trị tương ứng.

Nhiệm vụ: giúp hiểu một cách cơ bản về bảng băm, hàm băm.

Cách thực hiện:

- Xây dựng hàm băm.
- Xây dựng hàm khởi tạo các giá trị cho bảng băm sử dụng hàm băm ở trên.
- Nhập khoá cần tìm, cũng dùng hàm băm ở trên.

Code mẫu:

```
#include <stdio.h>

int Hash(int k, int M)
{
	if (M == 0)
		return 0;
	return (k % M);
}

void InitHash(int *&U, int M)
{
	int K[5] = {1,2,4,6,9};
	int i, pos;

	for(i = 0; i < M; i++)
		U[i] = 0;
	for(i = 0; i < 5; i++)
	{
		pos = Hash(K[i], M);
		U[pos] = K[i];
	}
}

int main(int argc, char* argv[])
{
	int M = 10;
	int *U = new int[M];
	int pos;

	InitHash(U, M);

	int x;
	printf("Nhap khoa tim kiem: ");
	scanf("%d", &x);

	pos = Hash(x, M);
	if (U[pos] == 0){
		printf("Khong tim thay khoa trong bang bam\n");
	}
	else{
		printf("Gia tri phan tu can tim kiem: %d\n", U[pos]);
	}

	return 0;
}
```

1. Biên dịch và chạy chương trình trên
2. Hãy chỉ ra công thức toán của hàm băm trong đoạn code mẫu trên
3. Trong hàm khởi tạo bảng băm, chỉ ra tập U có bao nhiêu phần tử, tập các khoá k lưu trong bảng băm có bao nhiêu phần tử.
4. Mô tả quy trình các bước chi tiết từ khi xây dựng bảng băm đến khi xuất ra kết quả tìm kiếm.

3.2 Hàm băm cho bài toán tra tên sinh viên

Sau khi đã nắm được những khái niệm về hàm băm, bảng băm và quy trình các bước chi tiết, sinh viên vận dụng vào bài toán tra tên SV theo MSSV.

Bài toán: Cho tập U các đối tượng sinh viên, thông tin mỗi SV bao gồm MSSV và tên SV đó. Giả sử ta có một tập k các SV trong Khoa CNTT, được lưu vào bảng băm U theo hàm băm modulo tương tự bài toán cơ bản. Hãy xây dựng ứng dụng cho phép tìm kiếm tên 1 SV theo MSSV được nhập từ bàn phím.

Cách thực hiện:

- Tạo cấu trúc Word biểu diễn cho thông tin của 1 SV, gồm hai trường: key (MSSV) và value (tên SV).
- Thực hiện hàm băm giống như bài toán cơ bản.
- Hàm khởi tạo bảng băm sẽ đưa danh sách các SV đang có vào bảng băm U.
- Khi tra cứu, băm MSSV của SV cần tra cứu, sau đó lấy thông tin của SV trong bảng băm U dựa theo giá trị vừa băm.

Code mẫu:

```
#include <stdio.h>
#include <string.h>

struct Word
{
	int key;
	char value[128];
};

int Hash(int k, int M)
{
	if (M == 0)
		return 0;
	return (k % M);
}

void InitHash(Word *&U, int M)
{
	Word K[5];
	K[0].key = 1;//MSSV
	strcpy(K[0].value, "Messi");
	K[1].key = 3;
	strcpy(K[1].value, "Ronaldo");
	K[2].key = 5;
	strcpy(K[2].value, "Rooney");
	K[3].key = 7;
	strcpy(K[3].value, "Drogba");
	K[4].key = 9;
	strcpy(K[4].value, "Xavi");

	int i, pos;

	for(i = 0; i < M; i++)
		U[i].key = 0;
	for(i = 0; i < 5; i++)
	{
		pos = Hash(K[i].key, M);
		U[pos] = K[i];
	}
}

int main(int argc, char* argv[])
{
```

```
	int M = 10;
	Word *U = new Word[M];
	int pos;

	InitHash(U, M);

	int x;
	printf("Nhap MSSV tim kiem: ");
	scanf("%d", &x);

	pos = Hash(x, M);
	if (U[pos].key == 0){
		printf("Khong tim thay SV nao trong bang bam\n");
	}
	else{
		printf("Ten cua SV can tim kiem la: %s\n", U[pos].value);
	}

	return 0;
}
```

1. Biên dịch đoạn chương trình trên.
2. Chỉ ra những thay đổi của chương trình này so với chương trình cơ bản.
3. Nếu bỏ đoạn code sau trong hàm băm

```
if (M == 0)
	return 0;
```

Thì có được không? Giải thích lý do.

4. Nếu thay K[i].key trong đoạn code sau

```
pos = Hash(K[i].key, M);
```

Thành

```
pos = Hash(K[i], M);
```

Thì chuyện gì xảy ra? Giải thích lý do tại sao.

5. Nếu thay giá trị K[2].key chỗ dòng code sau

```
K[2].key = 5;
```

Thành

```
K[2].key = 13;
```

Và lúc chạy, nhập MSSV là 3. Kết quả xuất ra là bao nhiêu? Đúng hay sai? Nếu sai, giải thích lý do tại sao?

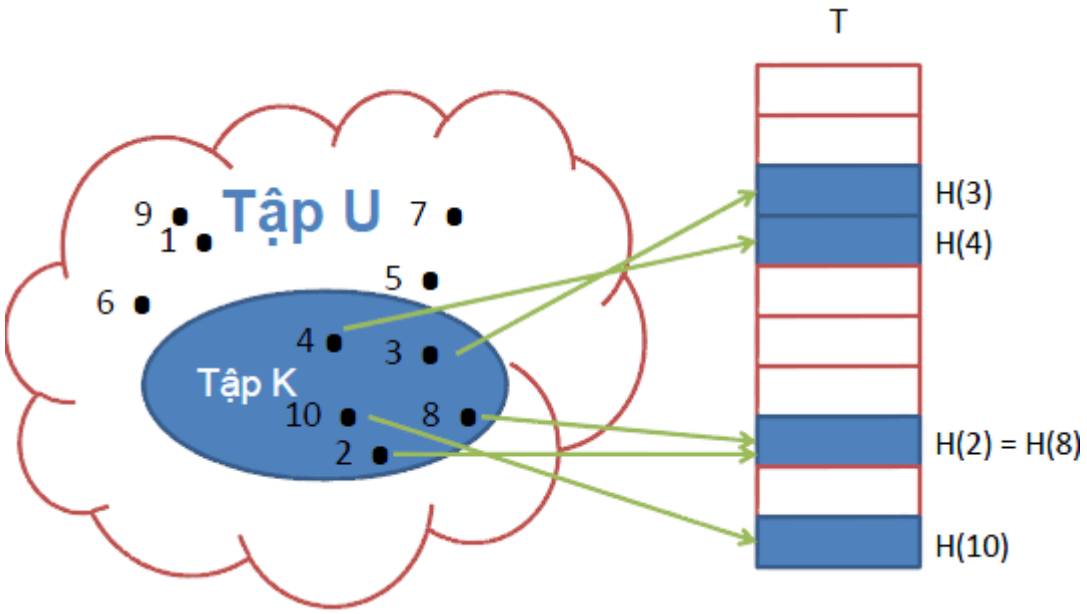
6. Viết lại chương trình trên, cho phép người dùng tự nhập số lượng giá trị k và thông tin của SV cần đưa vào bảng băm (ở đoạn code trên tác giả chỉ định cứng số lượng là 5).

4. GIẢI QUYẾT ĐỤNG ĐỘ

Sự đụng độ xảy ra khi tồn tại hai khoá có cùng giá trị băm.

$\exists k_1, k_2 \in K$:

$k_1 \neq k_2$, $\boldsymbol{H}(k_1) = \boldsymbol{H}(k_2)$



Các phương pháp giải quyết đụng độ:

- Phương pháp nối kết (chaining)
- Phương pháp địa chỉ mở (open-addressing)

Chi tiết các phương pháp xem trong phần lý thuyết.

Bài tập nâng cao:

1. Cải tiến bài toán tra thông tin SV theo tên của SV (giá trị băm là tên SV, giá trị xuất ra là thông tin SV như email, số điện thoại, MSSV…). Sử dụng hàm băm hợp lý để băm chuỗi tên SV.
   Gợi ý: hàm băm là hàm chuyển một chuỗi sang một số theo mã ASCII
2. Áp dụng phương pháp nối kết để giải quyết đụng độ cho bài toán tra tên SV.
3. Áp dụng phương pháp địa chỉ mở (dò tuyến tính) để giải quyết đụng độ cho bài toán tra tên SV.
4. Áp dụng phương pháp địa chỉ mở (dò bậc 2) để giải quyết đụng độ cho bài toán tra tên SV.
5. Áp dụng phương pháp địa chỉ mở (băm kép) để giải quyết đụng độ cho bài toán tra tên SV.