



# Hash Table

▼ MSSV	20280083
▼ Tên	Lại Toàn Thắng

## 3.1 Hàm băm cơ bản

1. Biên dịch và chạy chương trình trên
2. Hãy chỉ ra công thức toán của hàm băm trong đoạn code mẫu trên

Công thức toán trong hàm băm:

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

- Đây là phép ánh xạ
  - Công thức tính Modulo  $\rightarrow H(k) = k \bmod M$
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.
- U có  $M = 10$  phần tử
  - Tập các khoá k lưu trong bảng băm có 5 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.

```

24  int main(int argc, char* argv[])
25  {
26      int M = 10;
27      int* U = new int[M];
28      int pos;
29      InitHash(U, M);
30      int x;
31      printf("Nhap khoa tim kiem: ");
32      scanf("%d", &x);
33      pos = Hash(x, M);
34      if (U[pos] == 0) {
35          printf("Khong tim thay khoa trong bang bam\n");
36      }
37      else {
38          printf("Gia tri phan tu can tim kiem: %d\n", U[pos]);
39      }
40      return 0;
41  }

```

- Dòng 26 + 27:
  - InitHash: Khởi tạo bảng băm U gồm M = 10 phần tử
- U là tập hợp các giá trị khoá (value)
- K là tập hợp số khoá thực tế (key)
- $|U| > |K|$
- Dòng 29: InitHash
  - vòng lặp dòng 15 là khởi tạo mảng có giá trị các phần tử là 0
  - Dòng 7 là vòng lặp để quét các phần tử trong tập K
    - Hash từng phần tử trong K cho M để lấy index trong bảng băm U
    - Gán phần tử vừa hash vào bảng băm

```

11  void InitHash(int*& U, int M)
12  {
13      int K[5] = { 1,2,4,6,9 };
14      int i, pos;
15      for (i = 0; i < M; i++)
16          U[i] = 0;
17      for (i = 0; i < 5; i++)
18      {
19          pos = Hash(K[i], M);
20          U[pos] = K[i];
21      }
22  }

```

0	1	2	0	4	0	6	0	0	9
---	---	---	---	---	---	---	---	---	---

Đây là mảng U sau dòng 29 (InitHash)

- Từ dòng 30 trở đi là tìm kiếm phần tử dùng kĩ thuật Hashing tương tự như trên

## 3.2 Hàm băm cho bài toán tra tên sinh viên

1. Biên dịch đoạn chương trình trên.
2. Chỉ ra những thay đổi của chương trình này so với chương trình cơ bản.

Chương trình cơ bản

- Cấu trúc của tập K là int

Chương trình này

- Cấu trúc của tập K là Word (tức là có key và value)
  - Key theo kiểu int
  - Value theo kiểu char\*

3. Nếu bỏ đoạn code sau trong hàm băm

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

Thì có được không? Giải thích lý do.

Không được Vì M giống như cái size range của bảng băm U. Trường hợp mà  $M = 0$  thì bảng băm này gồm 0 phần tử thì không thể gọi là bảng băm được.

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

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.

Xuất hiện lỗi

sau:

```
cannot convert 'Word' to 'int'
```

vì hàm Hash (int k, int M) mà  $K[i]$  có kiểu dữ liệu là Word nên console hiện lỗi

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?

$K[2].key = 5$ :

```
Nhap MSSV tim kiem: 3
Ten cua SV can tim kiem la: Ronaldo
```

$K[2].key = 13$ :

```
Nhap MSSV tim kiem: 3
Ten cua SV can tim kiem la: Rooney
```

Kết quả xuất ra là đúng vì sau khi thay bằng 13. Thì ở bước Hash thì  $13 \bmod 10 = 3$  nên  $K[2].value = Rooney$  sẽ thay thế cho Ronaldo (tức  $K[2].value$  cũ)

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).

```

Word Input1sv()
{
    Word sv;
    cout << "Nhap ma so sinh vien = ";
    cin >> sv.key;
    cout << "Nhap ten sinh vien: ";
    cin.ignore();
    cin.getline(sv.value, 129);
    return sv;
}

```

```

void InitHash(Word*& U, int M)
{
    int n = 0;
    cout << "Nhap so luong sinh vien = ";
    cin >> n;
    Word K[n];
    for (int i = 0; i < n; i++)
    {
        K[i] = Input1sv();
    }
    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];
    }
}

```

#### 4. GIẢI QUYẾT ĐỤNG ĐỘ