

Lab 3: Bảng băm - Danh sách liên kết đôi

1 Bảng băm

1.1 Nội dung

Thông tin mã số thuế trong bài thực hành này được thu thập từ 1250 công ty Việt Nam. Thông tin chứa trong file *"MST.txt"*, chứa nội dung như sau:

```
1 Ten cong ty|MST|Địa chỉ
2 CONG TY TNHH BEE VIET NAM|0108927262|Số 8 - K8, Khu nhà ở liên kết trung tâm 75, Tổng cục II, Bộ Quốc Phòng, thôn Lai Xá, Xã Kim Chung, Huyện
Hoài Đức, Thành phố Hà Nội
3 CONG TY CO PHAN THUONG MAI CHAU DUC PHAT|3502406778|Số 266 Ấp Phước Trung, Xã Tam Phước, Huyện Long Điền, Tỉnh Bà Rịa - Vũng Tàu
4 CONG TY CO PHAN XAY DUNG VINH GIA PHAT|3603671412|Số 171, Xóm 4, Khu 2, Ấp Bàu Cá, Xã Trung Hòa, Huyện Trang Bom, Tỉnh Đồng Nai
5 CONG TY TNHH MTV THAN TAN HOANG LONG|0315938103|2/47 Đường Thanh Lọc 31, Khu Phố 3C, Phường Thanh Lọc, Quận 12, Thành phố Hồ Chí Minh
6 CONG TY TNHH NONG NGHIEP CONG NGHE CAO MIEN DONG VIET|3401194911|Thôn 5, Xã Tân Phúc, Huyện Hàm Tân, Tỉnh Bình Thuận
7 CONG TY TNHH XAY DUNG VINH GIA PHAT|3603671412|Số 171, Xóm 4, Khu 2, Ấp Bàu Cá, Xã Trung Hòa, Huyện Trang Bom, Tỉnh Đồng Nai
8 CONG TY TNHH THUONG MAI DICH VU PHU LONG RIVERSIDE|3401194929|243 Huỳnh Thúc Kháng, KP1, Phường Mũi Né, Thành phố Phan Thiết, Tỉnh Bình Thuận
9 CONG TY TNHH HANH TRANG PHAT|3603671155|To 5, Ấp Thanh Bình, Xã Lạc An, Huyện Long Thành, Tỉnh Đồng Nai
10 CONG TY TNHH THUONG MAI DICH VU THIET BI M.K.K|0315932380|154/1/34 Công Lộ, Phường 15, Quận Tân Bình, Thành phố Hồ Chí Minh
11 CONG TY TNHH TM - DV - NHA HANG HAI SAN KY QUANG|0315933352|Số 526 Đường Phạm Văn Dong, Phường 13, Quận Bình Thạnh, Thành phố Hồ Chí Minh
12 CONG TY TNHH MTV KIM LONG|1201613551|Số 170 Nguyễn Minh Duong, Ấp 1, Xã Đạo Thạnh, Thành phố Mỹ Tho, Tỉnh Tiền Giang
13 CONG TY TNHH SONA AGENCY VIET NAM|0108926660|Số 333 Bạch Mai, Phường Bạch Mai, Quận Hai Bà Trưng, Thành phố Hà Nội
```

trong đó:

- Dòng đầu tiên cung cấp các trường thông tin.
- Các dòng tiếp theo, mỗi dòng là thông tin của một công ty, được tách bởi một dấu "|".

Đối với bài thực hành này, sinh viên được yêu cầu đọc thông tin của các công ty từ tập tin *"MST.txt"* thành cấu trúc dữ liệu *Company*, và chứa trong một bảng băm.

1.2 Lập trình

Cấu trúc dữ liệu *Company* được định nghĩa như sau:

```
struct Company
{
    string name;
    string profit_tax;
    string address;
};
```

Hoàn thành các yêu cầu sau:

1. Viết hàm đọc thông tin các công ty từ tập tin cho trước:

- `vector<Company> readCompanyList(string file_name)`
- Đầu vào: `file_name` là tên tập tin đầu vào (*"MST.txt"*).
- Đầu ra: Danh sách các công ty được đọc từ tập tin và có kiểu dữ liệu là `vector<Company>`.

2. Viết hàm băm chuỗi (tên công ty):

- `long long hashString(string company_name)`
- Đầu vào: `company_name` là chuỗi (tên công ty) cần được băm.
- Đầu ra: `long long` - số nguyên dương, là kết quả của công thức băm được cho phía dưới.

- Công thức băm:

$$\text{hash}(s) = \left(\sum_{i=0}^{n-1} (s[i] \times p^i) \right) \bmod m$$

trong đó:

- s là 20 kí tự cuối cùng của chuỗi `company_name`. Nếu chuỗi không vượt quá 20 kí tự thì giữ nguyên.
- $s[i]$ là mã ASCII của kí tự tại vị trí thứ i trong s .
- $p = 31$
- $m = 10^9 + 9$

3. Viết hàm tạo một bảng băm kích thước 2000, được tạo từ danh sách các công ty:

- `Company* createHashTable(vector<Company> list_company)`
- Đầu vào: `list_company` - danh sách các công ty đọc từ tập tin.
- Đầu ra: Bảng băm được tạo.
- Lưu ý: Yêu cầu sử dụng kỹ thuật Dò tuyến tính.

4. Viết hàm thêm thông tin của một công ty vào bảng băm có sẵn:

- `void insert(Company* hash_table, Company company)`
- Đầu vào: - `hash_table` - bảng băm được cho.
- `company` - công ty cần được đưa vào bảng băm.

5. Hàm tìm kiếm công ty dựa vào tên công ty:

- `Company* search(Company* hash_table, string company_name)`
- Đầu vào: - `hash_table` - bảng băm được cho.
- `company_name` - tên của công ty cần tìm kiếm thông tin.
- Đầu ra: Công ty có tên được tìm kiếm, có kiểu dữ liệu là `Company*`. Trả về NULL nếu không tìm thấy tên công ty.

2 Danh sách liên kết đôi

Bạn được cho cấu trúc dữ liệu của một danh sách liên kết đôi như sau:

```
struct d_NODE{
    int key;
    d_NODE* pNext;
    d_NODE* pPrev;
};
```

```
struct d_List{
    d_NODE* pHead;
    d_NODE* pTail;
};
```

Cài đặt hàm thực hiện các chức năng sau:

1. Khởi tạo một danh sách từ số nguyên cho trước.
2. Thêm một node vào **đầu một danh sách** cho trước.
3. Thêm một node vào **trước một node** cho trước.
4. Thêm một node vào **sau một node** cho trước.
5. Thêm một node vào **cuối một danh sách** cho trước.
6. Xóa một node ở đầu một danh sách cho trước.
7. Xóa một node ở cuối một danh sách cho trước.
8. Xóa node đầu tiên có giá trị cho trước.