

Data Structures and Algorithms

HUS - HKII, 2023-2024

Assignment 8

Lecturer: Nguyễn Thị Tâm - Trần Bá Tuấn

§ Symbol Table and Hash Table §

Phần 1: Mục tiêu

- Sinh viên nắm được kiến thức về Symbol Table và triển khai được chương trình.
- Nắm được lý thuyết cơ bản về bảng băm (Hash Table), kỹ thuật xử lý đụng độ (va chạm). Hiểu được độ phức tạp của các thuật toán của bảng băm, các trường hợp sử dụng bảng băm và triển khai được HashTable.
- Áp dụng kiến thức về bảng băm và cấu trúc dữ liệu liên quan bảng băm để giải một số bài tập.

Phần 2: Tài liệu đọc thêm

- [Symbol Table](#)
- [Various implementations of Symbol Table](#)
- [Hash Table](#)
- [Hash Functions and list/types of Hash functions](#)
- [Separate Chaining Collision Handling Technique in Hashing](#)
- [Open Addressing Collision Handling technique in Hashing](#)

Phần 3: Thực hành

(1) Quy cách nộp bài

- Mỗi sinh viên hoàn thành bài tập trong package có tên Hw8_<idSinhvien>_<Hovaten>
- Trong đó, <idSinhvien> là mã sinh viên.
- Sinh viên nộp bài làm trên tài khoản của mình bao gồm:
 1. File nén .zip của thư mục chứa package (Hw8_<idSinhvien>_<Hovaten>.zip)
 2. Tất cả các file nguồn *.java.
- Sinh viên không nộp bài sẽ nhận điểm 0 bài tập tuần.
- Sinh viên CÓ GIAN LẬN trong nộp bài tập sẽ bị ĐÌNH CHỈ môn học (điểm 0 cho tất cả các điểm thành phần).

(2) Bài tập

Bài tập 1. Tạo giao diện cho cấu trúc dữ liệu **Bảng** như sau:

```

1 public interface SymbolTable<Key extends Comparable<Key>, Value> {
2     void put(Key key, Value value);
3     Value get(Key key);
4     void delete(Key key);
5     boolean contains(Key key);
6     boolean isEmpty();
7     int size();
8     Iterable<Key> keys();
9 }

```

Tạo lớp trừu tượng cài đặt giao diện trên như sau:

```

1 public abstract class AbstractSymbolTable<Key
2     extends Comparable<Key>, Value>
3     implements SymbolTable<Key, Value> {
4     public boolean contains(Key key) {
5         return get(key) != null;
6     }
7     public void delete(Key key) {
8         put(key, null);
9     }
10 }

```

Hai phương thức **contains** và **delete** được định nghĩa trước trong lớp trừu tượng này.

- Cài đặt cấu trúc dữ liệu **ArraySymbolTable** kế thừa lớp trừu tượng **AbstractSymbolTable** đã định nghĩa ở phần trên, sử dụng mảng không có thứ tự.
- Tạo giao diện **OrderedSymbolTable** kế thừa giao diện **SymbolTable** như sau:

```

1 public interface OrderedSymbolTable<Key extends Comparable<Key>, Value> extends
2     ↪ SymbolTable<Key, Value> {
3     Key min();
4     Key max();
5     Key floor(Key key);
6     Key ceiling(Key key);
7     int rank(Key key);
8     Key select(int k);
9     void deleteMin();
10    void deleteMax();
11    int size(Key u, Key v);
12    Iterable<Key> keys(Key u, Key v);
13 }

```

Cài đặt cấu trúc dữ liệu **OrderedArraySymbolTable** với **OrderedSymbolTable** sử dụng mảng có thứ tự. Sử dụng thuật toán tìm kiếm nhị phân cho việc tìm kiếm **key** trong bảng.

- Thực thi/ Cài đặt **SymbolTable** bằng cách sử dụng Danh sách liên kết (**LinkedList**) với gợi ý dưới đây:

class LinkedListSymbolTable extends AbstractSymbolTable<Key extends Comparable<Key>, Value>

Bài tập 2. Dựa vào kiến thức trên lớp và bài giảng, sinh viên hãy thực hiện các yêu cầu dưới đây:

- Tìm hiểu kiến thức về băm, các thành phần, như thế nào là hàm băm tốt, một số phương pháp thường gặp và cài đặt.
- Tìm hiểu về đụng độ (va chạm), khi nào xảy ra đụng độ (va chạm) và cách thức xử lý.
- Cài đặt giải pháp xử lý đụng độ với Seperate Chaining, Open Addressing (Linear Probing, Quadratic Probing and Double Probing).

Bài tập 3. Hoàn thiện các bài tập trong link sau: [Câu hỏi trắc nghiệm về cấu trúc dữ liệu Hash](#).

- **Yêu cầu:** Sinh viên tự chọn thực hiện tối thiểu 15 câu. Khi nộp bài, cần trình bày lại câu hỏi và câu trả lời. Nếu chọn câu trả lời nào thì cần trình bày giải thích chi tiết, rõ ràng. Quy tắc tên file cho bài là **Hw8_MaSinhvien_Hovaten_Baitap3.pdf**

Bài tập 4. Sinh viên thực hiện tối thiểu 2 bài trong link: [Luyện tập về Hash](#).

Bài tập 5. Sinh viên tự chọn thực hiện tối thiểu 3 bài trong link: [Luyện tập về Hash Fuction](#).