

Bài 0:

Tạo giao diện cho cấu trúc dữ liệu Bảng như sau:

```
public interface SimpleTable<Key extends Comparable<Key>, Value> {  
    void put(Key key, Value value);  
  
    Value get(Key key);  
  
    void delete(Key key);  
  
    boolean contains(Key key);  
  
    boolean isEmpty();  
  
    int size();  
  
    Iterable<Key> keys();  
}
```

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

```
public abstract class AbstractSimpleTable<Key extends Comparable<Key>, Value> implements SimpleTable<Key, Value> {  
    public boolean contains(Key key) {  
        return get(key) != null;  
    }  
  
    public void delete(Key key) {  
        put(key, null);  
    }  
}
```

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

Bài 1:

Cài đặt cấu trúc dữ liệu ArraySimpleTable kế thừa lớp trừu tượng AbstractSimpleTable đã định nghĩa ở bài 0, sử dụng mảng không có thứ tự.

Bài 2:

Cho mảng A đã được sắp xếp, viết chương trình cài đặt thuật toán tìm kiếm nhị phân phần tử x trên mảng A.

Bài 3:

Tạo giao diện `OrderedSimpleTable` kế thừa giao diện `SimpleTable` như sau:

```
public interface OrderedSimpleTable<Key extends Comparable<Key>,
Value> extends SimpleTable<Key, Value> {
    Key min();// Khóa nhỏ nhất

    Key max();// Khóa lớn nhất

    Key floor(Key key);// Khóa lớn nhất mà nhỏ hơn hoặc bằng key (cận dưới)

    Key ceiling(Key key);// Khóa nhỏ nhất mà lớn hơn hoặc bằng key (cận trên)

    int rank(Key key);// Số lượng khóa nhỏ hơn key

    Key select(int k);// Khóa tương ứng với rank(k)

    void deleteMin();//Xóa khóa nhỏ nhất

    void deleteMax();// xóa khóa lớn nhất

    int size(Key u, Key v);//số lượng key nằm trong khoảng từ u đến v

    Iterable<Key> keys(Key u, Key v);// Đưa ra các key có giá trị
    trong khoảng từ u đến v
}
```

Cài đặt cấu trúc dữ liệu `OrderedArraySimpleTable` cài đặt giao diện `OrderedSimpleTable`, sử dụng mảng có thứ tự. Sử dụng thuật toán tìm kiếm nhị phân cho việc tìm kiếm key trong bảng.