Data Structures and Algorithms

HUS HKI, 23 - 24

Assignment 3

Lecturer: Nguyễn Thị Hồng Minh
Trần Bá Tuấn - Đặng Trung Du

§ Abstract Data Type (ADT) and List ADT §

Phần 1: Mục tiêu

- Sinh viên cần nắm rõ kiểu dữ liệu trừu tượng đồng thời thực thi được các bài tập.
- Sinh viên hiểu rõ danh sách (List), đồng thời tạo danh sách bằng kiểu dữ liệu mảng (Array) và danh sách liên kết (LinkedList).
- Sinh viên cần nắm được định nghĩa, các thao tác và độ phức tạp của cấu trúc các danh sách liên kết phổ biến gồm danh sách liên kết đơn, danh sách liên kết đôi, danh sách liên kết vòng.

Phần 2: Thực hành

(1) Bài tập

Bài tập 1. *Tạo kiểu dữ liệu fraction number như sau:*

```
public class Fraction {
    private float numerator;
    private float denominator;
    public Fraction (float numerator, float denominator) {

    }
    public Fraction add(Fraction c) {
    }
    public Fraction minus(Fraction c) {
    }
    public Fraction multi(Fraction c) {
    }
    public Fraction divi(Fraction c) {
    }
    public Fraction normalize(Fraction c) {
    }
    public float getNumerator() {
        return numerator;
    }
    public float getDenominator() {
        return denominator;
    }
    @Override
    public String toString() {
    }
}
```

Sinh viên thực hiện các yêu cầu dưới đây:

- *Nhập dãy n phân số.*
- *In ra phân số ở vị trí thứ v trong dãy.*
- *Tính tổng của n phân số, cho kết quả là một phân số đã tối giản.*
- *Thực hiện tương tự với trừ, nhân và chia n phân số.*
- *(*) Tự đề xuất ít nhất một yêu cầu sử dụng kiểu phân số đã tạo ở trên, viết hàm thực hiện yêu cầu đó.*

Bài tập 2. *Tạo danh sách (List) bằng kiểu dữ liệu mảng (array) như sau:*

Tạo giao diện ListInterface kế thừa giao diện Iterable

```
public interface ListInterface<T> extends Iterable<T>{
    public void add(T data);
    public T get(int i);
    public void set(int i, T data);
    public void remove(T data);
    public void isContain(T data);
    public int size();
    public boolean isEmpty();
}
```

Tạo lớp SimpleArrayList cài đặt giao diện ListInterface

```
public class SimpleArrayList<T> implements ListInterface<T> {
    private T[] array;
    private int n = 0;
    private int defaultSize = 100;

    public SimpleArrayList() {
        array = (T[]) new Object[defaultSize];
    }
    public SimpleArrayList(int capacity) {

    }
    public void add(T data) {

    }
    public T get(int i) {

    }
    public void set(int i, T data) {

    }
    public void remove(T data) {

    }
    public boolean isContain(T data) {

    }
    public int size() {

    }
    public boolean isEmpty() {
    }
    public Iterator<T> iterator() {

    }
}
```

Yêu cầu: *Sinh viên hoàn thiện thân hàm của các hàm trên.*

Bài tập 3. *Tạo kiểu danh sách liên kết đơn (SimpleLinkedList) theo mẫu sau.*

```
public class SimpleLinkedList<T> {
    class Node {
        T data;
        Node next;
    }
    private Node top = null;
    private Node bot = null;
    private int n = 0;
    public void add(T data) {

    }
    public void addBot(T data) {

    }
    public T get(int i) {
        return null;
        }
    public void set(int i, T data) {
    }
    public boolean isContain(T data) {
        return false;
    }
    public int size() {
        return 0;
    }
    public boolean isEmpty() {
        return true;
    }
    public T removeTop() {
        return null;
    }
    public T removeBot() {
        return null;
    }
    public void remove(T data) {
    }
}
```

Bài tập 4. *(*) Sử dụng các cấu trúc dữ liệu ở bài 2, 3 để viết chương trình đếm số lần xuất hiện của các từ trong một văn bản cho trước nhập vào từ bàn phím hoặc từ file văn bản.*

Tạo đối tượng là WordCount gồm 2 thuộc tính là word và count. Đối tượng WordCount nạp chồng phương thức equals(Object o) để có thể sử dụng phương thức isContain đã xây dựng ở các cấu trúc dữ liệu trên, hoặc có thể sử dụng phương thức indexOf của các đối tượng cài đặt giao diện List.

(2) Bài tập thêm

MỤC TIÊU

- Giúp sinh viên có thêm nhiều bài tập để luyện tập khả năng lập trình của bản thân.
- Nắm rõ kiến thức, thành thạo cài đặt về danh sách liên kết đơn, danh sách liên kết đôi và danh sách liên kết vòng.

DANH SÁCH LIÊN KẾT ĐƠN

Bài tập 5. *Sinh viên tự chọn 3 bài trong Bài tập từ 21 đến 31 tại link:*

https://codelearn.io/learning/cau-truc-du-lieu-va-giai-thuat

Bài tập 6. ***Mô tả yêu cầu:***

- *Cài đặt danh sách liên kết đơn.*
- *Thực hành các thao tác cơ bản, khởi tạo danh sách liên kết đơn, các thao tác chèn, xóa.*

https://leetcode.com/problems/design-linked-list/

Bài tập 7. ***Số phần tử trong danh sách liên kết***

https://practice.geeksforgeeks.org/problems/count-nodes-of-linked-list/1

- *Nguồn: https://www.geeksforgeeks.org/find-length-of-a-linked-list-iterative-and-recursive/*
- ***Gợi ý:*** *Duyệt danh sách liên kết, sau đó đếm số phần tử.*

Bài tập 8. ***Phần tử thứ*** N ***từ cuối danh sách liên kết***

https://practice.geeksforgeeks.org/problems/nth-node-from-end-of-linked-list/1

- *Nguồn: https://www.geeksforgeeks.org/nth-node-from-the-end-of-a-linked-list/*
- ***Gợi ý:*** *Duyệt danh sách liên kết, đếm* M *là số phần tử, sau đó duyệt danh sách liên kết lần nữa để tìm đến phần tử thứ* $M - N + 1$.

Bài tập 9. ***Đếm số lần xuất hiện của một giá trị trong danh sách liên kết***

https://practice.geeksforgeeks.org/problems/occurence-of-an-integer-in-a-linked-list/1

- ***Nguồn:*** *https://www.geeksforgeeks.org/write-a-function-that-counts-the-number-of-times-a-given-int-occurs-in-a-linked-list/*
- ***Gợi ý:*** *Duyệt danh sách liên kết.*

Bài tập 10. ***Đảo ngược các phần tử trong danh sách liên kết***

https://www.geeksforgeeks.org/reverse-a-linked-list/

- ***Nguồn:*** *https://www.geeksforgeeks.org/print-reverse-of-a-linked-list-without-actually-reversing/*
- ***Gợi ý:*** *Tương đương với việc tạo một danh sách liên kết mới thông qua duyệt danh sách liên kết hiện tại và thêm mỗi phần tử của danh sách hiện tại vào đầu danh sách mới.*

Bài tập 11. ***Kiểm tranh danh sách liên kết có là đảo ngược (palindrome) hay không?***

https://leetcode.com/problems/palindrome-linked-list/

- ***Nguồn:*** *https://www.geeksforgeeks.org/function-to-check-if-a-singly-linked-list-is-palindrome/*
- ***Gợi ý:*** *Tương đương với việc tạo danh sách liên kết đảo ngược và sau đó duyệt đồng thời hai danh sách để kiểm tra hai danh sách có chứa cùng nội dung.*

Bài tập 12. ***Xóa các phần tử trùng nhau trong danh sách không giảm***

https://leetcode.com/problems/remove-duplicates-from-sorted-list/

- ***Gợi ý:*** *So sánh hai phần tử trong danh sách liên kết, nếu có cùng giá trị thì xóa phần tử đứng sau.*

Bài tập 13. ***Cộng hai đa thức được biểu diễn bằng danh sách liên kết***

https://practice.geeksforgeeks.org/problems/polynomial-addition/1

- ***Nguồn:*** *https://www.geeksforgeeks.org/adding-two-polynomials-using-linked-list/*

DANH SÁCH LIÊN KẾT ĐÔI

Bài tập 14. ***Thêm phần tử mới trong danh sách liên kết đôi***

https://practice.geeksforgeeks.org/problems/insert-a-node-in-doubly-linked-list/1

- ***Nguồn:*** *https://www.geeksforgeeks.org/doubly-linked-list/*

Một số yêu cầu khác:

- ***Xóa phần tử trong danh sách liên kết đôi:*** *https://practice.geeksforgeeks.org/problems/delete-node-in-doubly-linked-list/1*
  Nguồn: *https://www.geeksforgeeks.org/delete-a-node-in-a-doubly-linked-list/*
- ***Chèn phần tử vào danh sách liên kết đôi tăng dần*** *https://practice.geeksforgeeks.org/problems/insert-in-sorted-way-in-a-sorted-dll/1*
  Nguồn: *https://www.geeksforgeeks.org/insert-value-sorted-way-sorted-doubly-linked-list/*

Bài tập 15. ***Quay danh sách liên kết đôi*** k ***lần***

https://practice.geeksforgeeks.org/problems/rotate-doubly-linked-list-by-p-nodes/1

- ***Nguồn:*** *https://www.geeksforgeeks.org/rotate-doubly-linked-list-n-nodes/*

DANH SÁCH LIÊN KẾT VÒNG

Bài tập 16. ***Duyệt danh sách liên kết vòng***

https://practice.geeksforgeeks.org/problems/circular-linked-list-traversal/1

- ***Nguồn:*** *https://www.geeksforgeeks.org/circular-linked-list-set-2-traversal/*

Một số yêu cầu khác:

- ***Kiểm tra một danh sách có là danh sách liên kết vòng không***

  https://practice.geeksforgeeks.org/problems/circular-linked-list/1

  Nguồn: *https://www.geeksforgeeks.org/check-if-a-linked-list-is-circular-linked-list/*
- ***Thêm phần tử vào danh sách liên kết vòng tăng dần***

  https://practice.geeksforgeeks.org/problems/sorted-insert-for-circular-linked-list/1

  Nguồn: *https://www.geeksforgeeks.org/sorted-insert-for-circular-linked-list/*
- ***Xóa và đảo ngược danh sách liên kết vòng***

  https://practice.geeksforgeeks.org/problems/deletion-and-reverse-in-linked-list/1

  Nguồn: *https://www.geeksforgeeks.org/deletion-circular-linked-list/*

Bài tập 17. ***Tách danh sách liên kết vòng ra làm hai danh sách liên kết vòng***

https://practice.geeksforgeeks.org/problems/split-a-circular-linked-list-into-two-halves/1

- ***Nguồn:*** *https://www.geeksforgeeks.org/split-a-circular-linked-list-into-two-halves/*

(3) Quy cách nộp bài

- Mỗi sinh viên hoàn thành bài tập trong package có tên Hw3_<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 (Hw3_<idSinhvien>_<Hovaten>.zip)
  2. Tất cả các file nguồn *.java.