

## BÀI TẬP CHƯƠNG 1:

**Bài 1.1:** Viết hàm đệ quy tính  $n!$

**Bài 1.2:** Dãy số Fibonacci  $\{F_n\}$  được xác định như sau:  $F_0=1$ ,  $F_1=1$ ,  $F_n=F_{n-1} + F_{n-2}$  với  $N \geq 3$ .

a. Viết hàm đệ quy  $Fib(n)$  để tính  $F_n$

b. Viết hàm lặp  $Fib(n)$  để tính  $F_n$

**Bài 1.3:** Một chuỗi ký tự được gọi là Palindrome nếu đảo ngược các ký tự của nó ta nhận được chuỗi ban đầu, ví dụ từ M D M là Palindrome. Trình bày thuật toán kiểm tra một xâu ký tự có phải là Palindrome hay không?

**Bài 1.4:** Hai chuỗi ký tự là Anagram của nhau nếu một chuỗi là hoán vị các ký tự của chuỗi kia, ví dụ “dear” và “read” là Anagram của nhau. Viết chương trình nhập hai chuỗi, rồi hiển thị một trong các thông báo sau: “Là Anagram”, “Không là Anagram”.

**Bài 1.5:** Số hoàn thiện  $n$  là số tự nhiên có tổng các ước số thực sự (kể cả 1) bằng chính nó. Ví dụ  $n=6$  là số hoàn thiện vì  $6 = 1 + 2 + 3$ . Trình bày thuật toán kiểm tra xem một số tự nhiên có phải là số hoàn thiện hay không?

**Bài 1.6:** Bài toán Tháp Hà Nội: Có 3 cọc, cọc thứ nhất có  $n$  đĩa kích thước khác nhau xếp chồng nhau, đĩa nhỏ ở trên đĩa lớn ở dưới. Hãy chuyển các đĩa từ cọc thứ nhất sang cọc thứ hai, cho sử dụng cọc trung gian thứ ba, sao cho luôn đảm bảo đĩa nhỏ ở trên đĩa lớn ở dưới và mỗi lần chỉ được chuyển 1 đĩa. Viết hàm đệ quy tìm phương án di chuyển đĩa tối ưu.

## BÀI TẬP LÝ THUYẾT CHƯƠNG 2:

**Bài 2.1:** Viết một chương trình C hoàn chỉnh thực hiện nhiều lần việc tạo ra bảng thực đơn, rồi lựa chọn để thực hiện các công việc sau đây:

1. Nhập danh sách đặc có tối đa 100 sinh viên gồm họ tên, chiều cao, rồi tính cân nặng tiêu chuẩn.
2. Liệt kê các phần tử trong danh sách.
3. Nhập họ tên, chiều cao, tính cân nặng tiêu chuẩn, và nhập vị trí để thêm người đó vào danh sách.
4. Nhập một họ tên từ bàn phím, rồi tìm và liệt kê phần tử đầu tiên có họ tên đó (nếu có).
5. Nhập một vị trí rồi xóa phần tử có vị trí đó ra khỏi danh sách.
6. Nhập một họ tên, rồi tìm và xóa người đầu tiên có họ tên đó ra khỏi danh sách (nếu có).

**Bài 2.2:** Viết chương trình C hoàn chỉnh thực hiện nhiều lần việc tạo bảng thực đơn rồi lựa chọn để thực hiện các công việc sau đây:

1. Nhập mới từ bàn phím danh sách liên kết các sinh viên gồm họ tên, chiều cao, rồi tính cân nặng tiêu chuẩn theo công thức:

$$\text{Cân nặng tiêu chuẩn} = \text{Chiều cao} * 100 - 105$$

Yêu cầu viết chương trình nhập sao cho kết quả ta nhận được danh sách liên kết gồm các phần tử có thứ tự ngược với các phần tử đã nhập (muốn vậy ban đầu ta cho danh sách rỗng, và mỗi lần thêm 1 phần tử mới thì ta luôn luôn thêm nó vào đầu danh sách).

2. Liệt kê các phần tử trong danh sách.

Viết thêm các yêu cầu:

3. Thêm một người tên LÊ THÊM, cao 1.7 vào đầu danh sách (có tính cân nặng tiêu chuẩn).
4. Nhập một họ tên, chiều cao, rồi tính cân nặng tiêu chuẩn và thêm người này vào đầu danh sách.

Viết thêm các yêu cầu:

5. Tìm xem có ai tên LÊ TÌM không, nếu có thì liệt kê các thông tin của người đầu tiên tìm được, còn không thì thông báo là không có.
6. Nhập một họ tên từ bàn phím, rồi tìm và liệt kê các thông tin của người đầu tiên tìm được có họ tên đó (nếu có).
7. Xóa người đầu tiên trong danh sách.
8. Nhập một họ tên từ bàn phím, rồi tìm và xóa người đầu tiên có họ tên đó (nếu có).
9. Viết hàm con tính tổng chiều cao.
10. Viết hàm con tính chiều cao lớn nhất (dễ hơn).
11. Viết hàm con tính chiều cao nhỏ nhất (khó hơn).
12. Viết hàm con đếm số lượng người có chiều cao trên 1.55m.

**Bài 2.3:** Phân tích ưu, khuyết điểm của xâu liên kết so với mảng. Tổng quát hóa các trường hợp nên dùng xâu liên kết.

**Bài 2.4:** Xây dựng một cấu trúc dữ liệu thích hợp để biểu diễn đa thức  $P(x)$  có dạng :

$$P(x) = c_1x^n + c_2x^{n-1} + \dots + c_kx^k$$

Biết rằng:

- Các thao tác xử lý trên đa thức bao gồm :
  - + Thêm một phần tử vào cuối đa thức.

+ In danh sách các phần tử trong đa thức theo :

- . thứ tự nhập vào.
- . ngược với thứ tự nhập vào.

+ Hủy một phần tử bất kỳ trong danh sách.

- Số lượng các phần tử không hạn chế.
- Chỉ có nhu cầu xử lý đa thức trong bộ nhớ chính.

a. Giải thích lý do chọn CTDL đã định nghĩa.

b. Viết chương trình con ước lượng giá trị của đa thức  $P(x)$  khi biết  $x$ .

c. Viết chương trình con rút gọn biểu thức (gộp các phần tử cùng số mũ).

**Bài 2.5:** Xét đoạn chương trình tạo một xâu đơn gồm 4 phần tử (không quan tâm dữ liệu) sau đây:

```
Dx=NULL;p=Dx;
```

```
Dx=new (NODE);
```

```
for(i=0; i < 4; i++)
```

```
{    p = p->next;
```

```
    p = new (NODE);
```

```
}
```

```
(*p).next = NULL;
```

Đoạn chương trình có thực hiện được thao tác tạo nêu trên không ? Tại sao ? Nếu không thì có thể sửa lại như thế nào cho đúng ?

**Bài 2.6:** Một ma trận chỉ chứa rất ít phần tử với giá trị có nghĩa (ví dụ: phần tử khác không) được gọi là ma trận thưa. Dùng cấu trúc xâu liên kết để tổ chức biểu diễn một ma trận thưa sao cho tiết kiệm nhất (chỉ lưu trữ các phần tử có nghĩa).

a)Viết chương trình cho phép nhập, xuất ma trận.

b)Viết chương trình con cho phép cộng hai ma trận.

**Bài 2.7:** Bài toán Josephus : có N người đã quyết định tự sát tập thể bằng cách đứng trong vòng tròn và giết người thứ M quanh vòng tròn, thu hẹp hàng ngũ lại khi từng người lần lượt ngã khỏi vòng tròn. Vấn đề là tìm ra thứ tự từng người bị giết.

Ví dụ : N = 9, M = 5 thì thứ tự là 5, 1, 7, 4, 3, 6, 9, 2, 8

Hãy viết chương trình giải quyết bài toán Josephus, xử dụng cấu trúc xâu liên kết.

**Bài 2.8:** Hãy cho biết nội dung của stack sau mỗi thao tác trong dãy :

EAS\*Y\*\*QUE\*\*\*ST\*\*\*I\*ON

Với một chữ cái tượng trưng cho thao tác thêm chữ cái tương ứng vào stack, dấu \* tượng trưng cho thao tác lấy nội dung một phần tử trong stack in lên màn hình.

Hãy cho biết sau khi hoàn tất chuỗi thao tác, những gì xuất hiện trên màn hình ?

**Bài 2.9:** Hãy cho biết nội dung của hàng đợi sau mỗi thao tác trong dãy :

EAS\*Y\*\*QUE\*\*\*ST\*\*\*I\*ON

Với một chữ cái tượng trưng cho thao tác thêm chữ cái tương ứng vào hàng đợi, dấu \* tượng trưng cho thao tác lấy nội dung một phần tử trong hàng đợi in lên màn hình.

Hãy cho biết sau khi hoàn tất chuỗi thao tác, những gì xuất hiện trên màn hình ?

**Bài 2.10:** Giả sử phải xây dựng một chương trình soạn thảo văn bản, hãy chọn cấu trúc dữ liệu thích hợp để lưu trữ văn bản trong quá trình soạn thảo. Biết rằng :

- Số dòng văn bản không hạn chế.
- Mỗi dòng văn bản có chiều dài tối đa 80 ký tự.
- Các thao tác yêu cầu gồm :
  - + Di chuyển trong văn bản (lên, xuống, qua trái, qua phải)
  - + Thêm, xoá sửa ký tự trong một dòng
  - + Thêm, xoá một dòng trong văn bản
  - + Đánh dấu, sao chép khối

Giải thích lý do chọn cấu trúc dữ liệu đó.

**Bài 2.11:** Viết hàm ghép 2 xâu vòng  $L_1$ ,  $L_2$  thành một xâu vòng L với phần tử đầu xâu là phần tử đầu xâu của  $L_1$ .

## BÀI TẬP THỰC HÀNH CHƯƠNG 2:

**Bài 2.12:** Cài đặt thuật toán sắp xếp Chèn trực tiếp trên xâu kép. Có phát huy ưu thế của thuật toán hơn trên mảng hay không ?

**Bài 2.13:** Cài đặt thuật toán QuickSort theo kiểu không đệ qui.

**Bài 2.14:** Cài đặt thuật toán MergeSort trên xâu kép.

**Bài 2.15:** Cài đặt lại chương trình quản lý nhân viên theo bài tập 1.6 chương 1, nhưng sử dụng cấu trúc dữ liệu xâu liên kết. Biết rằng số nhân viên không hạn chế.

**Bài 2.17:** Cài đặt một chương trình soạn thảo văn bản theo mô tả trong bài tập 2.8.

**Bài 2.18:** Cài đặt chương trình tạo một bảng tính cho phép thực hiện các phép tính +, -, \*, /, div trên các số có tối đa 30 chữ số, có chức năng nhớ (M+, M-, MC, MR).

**Bài 2.19:** Cài đặt chương trình cho phép nhập vào một biểu thức gồm các số, các toán tử +, -, \*, /, %, các hàm toán học sin, cos, tan, ln, ex, dấu mở, đóng ngoặc "(", ") " và tính toán giá trị của biểu thức này.

---o-O-o---

### BÀI TẬP LÝ THUYẾT CHƯƠNG 3:

**Bài 3.1.** Hãy trình bày các vấn đề sau đây:

- Định nghĩa và đặc điểm của cây nhị phân tìm kiếm.
- Thao tác nào thực hiện tốt trong kiểu này.
- Hạn chế của kiểu này là gì ?

**Bài 3.2.** Xét thuật giải tạo cây nhị phân tìm kiếm. Nếu thứ tự các khóa nhập vào là như sau:

8 3 5 2 20 11 30 9 18 4

thì hình ảnh cây tạo được như thế nào ?

Sau đó, nếu hủy lần lượt các nút theo thứ tự như sau :

15, 20

thì cây sẽ thay đổi như thế nào trong từng bước hủy, vẽ sơ đồ (nêu rõ phương pháp hủy khi nút có cả 2 cây con trái và phải)

**Bài 3.3.** Áp dụng thuật giải tạo cây nhị phân tìm kiếm cân bằng để tạo cây với thứ tự các khóa nhập vào là như sau :

5 7 2 1 3 6 10

thì hình ảnh cây tạo được như thế nào ? Giải thích rõ từng tình huống xảy ra khi thêm từng khóa vào cây và vẽ hình minh họa.

Sau đó, nếu hủy lần lượt các nút theo thứ tự như sau :

5, 6, 7, 10

thì cây sẽ thay đổi như thế nào trong từng bước hủy, vẽ sơ đồ và giải thích

**Bài 3.4.** Viết các hàm xác định các thông tin của cây nhị phân T:

- Số nút lá
- Số nút có đúng 1 cây con
- Số nút có đúng 2 cây con
- Số nút có khóa nhỏ hơn x (giả sử T là CNPTK)
- Số nút có khóa lớn hơn x (giả sử T là CNPTK)
- Số nút có khóa lớn hơn x và nhỏ hơn y (T là CNPTK)
- Chiều cao của cây
- In ra tất cả các nút ở tầng (mức) thứ k của cây T
- In ra tất cả các nút theo thứ tự từ tầng 0 đến tầng thứ h-1 của cây T (h là chiều cao của T).
- Kiểm tra xem T có phải là cây cân bằng hoàn toàn không.
- Độ lệch lớn nhất trên cây. (Độ lệch của một nút là độ lệch giữa chiều cao của cây con trái và cây con phải của nó. Độ lệch lớn nhất trên cây là độ lệch của nút có độ lệch lớn nhất).

**Bài 3.5.** Xây dựng cấu trúc dữ liệu biểu diễn cây N-phân ( $2 < N \leq 20$ ).

- Viết chương trình con duyệt cây N-phân và tạo sinh cây nhị phân tương ứng với các khoá của cây N-phân.
- Giả sử khóa được lưu trữ chiếm k byte, mỗi con trỏ chiếm 4 byte, vậy dùng cây nhị phân thay cây N-phân thì có lợi gì trong việc lưu trữ các khoá ?

**Bài 3.6.** Viết hàm chuyển một cây N-phân thành cây nhị phân.

**Bài 3.7.** Viết hàm chuyển một cây nhị phân tìm kiếm thành xâu kép có thứ tự tăng dần.

**Bài 3.8.** Giả sử A là một mảng các số thực đã có thứ tự tăng. Hãy viết hàm tạo một cây nhị phân tìm kiếm có chiều cao thấp nhất từ các phần tử của A.

**Bài 3.9.** Viết chương trình con đảo nhánh (nhánh trái của một nút trên cây trở thành nhánh phải của nút đó và ngược lại) một cây nhị phân.

**Bài 3.10.** Hãy vẽ cây AVL với 12 nút có chiều cao cực đại trong tất cả các cây AVL 12 nút.

**Bài 3.11.** Tìm một dãy N khóa sao cho khi lần lượt dùng thuật toán thêm vào cây AVL để xen các khóa này vào cây sẽ phải thực hiện mỗi thao tác cân bằng lại (LL, LR, RL, RR) ít nhất một lần.

**Bài 3.12.** Hãy tìm một ví dụ về một cây AVL có chiều cao là 6 và khi hủy một nút lá (chỉ ra cụ thể) việc cân bằng lại lan truyền lên tận gốc của cây. Vẽ ra từng bước của quá trình hủy và cân bằng lại này.

### BÀI TẬP THỰC HÀNH CHƯƠNG 3:

**Bài 3.13.** Cài đặt chương trình mô phỏng trực quan các thao tác trên cây nhị phân tìm kiếm.

**Bài 3.14.** Cài đặt chương trình mô phỏng trực quan các thao tác trên cây AVL.

**Bài 3.15.** Viết chương trình cho phép tạo, tra cứu và sửa chữa từ điển Anh- Việt.

**Bài 3.16.** Viết chương trình khảo sát tần xuất xảy ra việc cân bằng lại của các thao tác thêm và hủy một phần tử trên cây AVL bằng thực nghiệm. Chương trình này phải cho phép tạo lập ngẫu nhiên các cây AVL và xóa ngẫu nhiên cho đến khi cây rỗng. Qua đó cho biết số lần xảy ra cân bằng lại trung bình của từng thao tác.

### BÀI TẬP LÝ THUYẾT CHƯƠNG 4:

**Bài 4.1.** Trong 3 phương pháp sắp xếp cơ bản (chọn trực tiếp, chèn trực tiếp, nổi bọt) phương pháp nào thực hiện sắp xếp nhanh nhất với một dãy đã có thứ tự? Giải thích.

**Bài 4.2.** Cho một ví dụ minh họa ưu điểm của thuật toán ShakeSort đối với BubbleSort khi sắp xếp một dãy số.

**Bài 4.3.** Cho dãy số 5 1 2 8 4 7 0 12 4 3 24 1 4, hãy minh họa kết quả sắp xếp dãy số này từng bước với các giải thuật chọn trực tiếp, chèn trực tiếp, nổi bọt.

### BÀI TẬP THỰC HÀNH CHƯƠNG 4:

**Bài 4.4.** Cài đặt các thuật toán sắp xếp đã trình bày. Thử nghiệm trực quan các thao tác của thuật toán. Tính thời gian thực hiện của mỗi thuật toán.

**Bài 4.5** Cài đặt thêm chức năng xuất bảng lương nhân viên theo thứ tự tiền lương tăng dần cho bài tập.

### BÀI TẬP LÝ THUYẾT CHƯƠNG 5:

**Bài 5.1.** Xét mảng các số nguyên có nội dung như sau :

-9      -9      -5      -2      0      3      7      7      10      15

a. Tính số lần so sánh để tìm ra phần tử  $X = -9$  bằng phương pháp:

- Tìm tuyến tính
- Tìm nhị phân

Nhận xét và so sánh 2 phương pháp tìm nêu trên trong trường hợp này và trong trường hợp tổng quát.

b. Trong trường hợp tìm nhị phân, phần tử nào sẽ được tìm thấy (thứ 1 hay 2) .

**Bài 5.2.** Xây dựng thuật toán tìm phần tử nhỏ nhất (lớn nhất) trong một mảng các số nguyên.

### BÀI TẬP THỰC HÀNH CHƯƠNG 5:

**Bài 5.3.** Cài đặt các thuật toán tìm kiếm đã trình bày. Thử nghiệm trực quan các thao tác của thuật toán. Tính thời gian thực hiện của mỗi thuật toán.

**Bài 5.4.** Hãy viết hàm tìm tất cả các số nguyên tố nằm trong mảng một chiều a có n phần tử.

**Bài 5.5.** Hãy viết hàm tìm dãy con tăng dài nhất của mảng một chiều a có n phần tử (dãy con là một dãy liên tiếp các phần của a).

**Bài 5.6.** Cài đặt thuật toán tìm phần tử trung vị (median) của một dãy số.

### BÀI TẬP CHƯƠNG 6:

**Bài 6.1.** Xây dựng bảng băm quản lý các sinh viên gồm các thông tin: Họ tên, năm sinh, điểm tin, điểm anh văn, điểm trung bình, xếp loại.

Bảng băm gồm 5 lớp chứa các sinh viên theo xếp loại:

- Lớp 0 chứa các sinh viên xếp loại kém (0 ≤ Điểm trung bình < 4).
- Lớp 1 chứa các sinh viên xếp loại yếu (4 ≤ Điểm trung bình < 5).
- Lớp 2 chứa các sinh viên xếp loại trung bình (5 ≤ Điểm trung bình < 6.5).
- Lớp 3 chứa các sinh viên xếp loại khá (6.5 ≤ Điểm trung bình < 8).
- Lớp 4 chứa các sinh viên xếp loại giỏi (8 ≤ Điểm trung bình ≤ 10).

Nhập danh sách các sinh viên từ bàn phím gồm họ tên, năm sinh, điểm tin, điểm anh văn, rồi tính điểm trung bình biết rằng môn tin có hệ số 2, rồi tính xếp loại, và lưu vào bảng băm.

Liệt kê danh sách các sinh viên của từng lớp trong bảng băm.

---o-O-o---