

BÀI TẬP TUẦN 2

Môn học: Cấu Trúc Dữ Liệu và Giải Thuật.

GV hướng dẫn thực hành: Nguyễn Khánh Toàn (ktoan271199@gmail.com).

Nội dung chính:

- Ôn tập cấu trúc dữ liệu danh sách liên kết.
- Ôn tập đệ quy

Thời gian thực hiện: 2 tuần.

Bài tập được biên soạn lại và tham khảo từ tài liệu thực hành môn Cấu Trúc Dữ Liệu và Giải Thuật của bộ môn Công Nghệ Tri Thức, trường Đại Học Khoa Học Tự Nhiên TP HCM. Trân trọng cảm ơn quý Thầy Cô của bộ môn.

1 Đệ quy

Các bạn sinh viên giải và cài đặt các bài toán sau (sử dụng đệ quy).

1. Nhập đầu vào là số nguyên dương n .

Tính giá trị biểu thức sau: $S = 1 + 2 + 3 + \dots + n$

2. Viết hàm đệ quy tính giá trị số Fibonacci thứ n ($n \geq 0$). Số Fibonacci được định nghĩa như sau:

- $F_0 = 0, F_1 = 1$
- $F_n = F_{n-1} + F_{n-2}, n \geq 2$

3. Nhập input đầu vào là hai số nguyên a, b . Các bạn viết hàm tìm ước chung lớn nhất của hai số này sử dụng đệ quy.

Ví dụ: Nhập "4, 6", ước chung lớn nhất của hai số này trả ra là 2.

4. Cho trước một mảng động đã bao gồm giá trị, và các giá trị đó được sắp xếp theo thứ tự tăng dần. Các bạn viết hàm tìm kiếm phần tử có giá trị x trong mảng sử dụng thuật toán tìm kiếm nhị phân được viết dưới dạng đệ quy.

int recursiveBinarySearch(**int*** a, **int** left, **int** right, **int** key);

5. Cho 5 ký tự "A, B, C, D, E". Các bạn viết hàm đệ quy in ra các hoán vị được tạo thành từ toàn bộ 5 ký tự này, ví dụ *ABCDE, ACDEB, ...*

2 Danh sách liên kết

2.1 Cài đặt các hàm quan trọng của DSLK

Các bạn sinh viên thiết kế cách tổ chức cấu trúc dữ liệu danh sách liên kết và thực hiện cài đặt các hàm sau:

1. Thiết kế danh sách liên kết đơn

```
struct Node{
    int key;
    Node* p_next;
};

struct SinglyLinkedList{
    Node* p_head;
    Node* p_tail;
};
```

Hình 1: Danh sách liên kết đơn

2. Khởi tạo một nút với giá trị cho trước.
Node* initializeNode(**int** key);
 3. Khởi tạo một danh sách liên kết từ một nút cho trước.
void initializeList(**List*** &L, **Node*** &head)
 4. Thêm một phần tử vào đầu danh sách liên kết (hãy khởi tạo một nút mới chứa giá trị đó).
void addElement(**List*** &L, **int** value);
 5. Xóa nút đầu tiên khỏi danh sách liên kết.
void removeElement(**List*** &L);
 6. Xóa tất cả các nút khỏi danh sách liên kết.
void removeAllElement(**List*** &L);
 7. Thêm một phần tử vào vị trí thứ x của danh sách liên kết (hãy khởi tạo một nút mới chứa giá trị đó).
void addElement(**List*** &L, **int** index, **int** value);
 8. Xóa nút có vị trí thứ x khỏi danh sách liên kết.
void removeElement(**List*** &L, **int** index);
 9. Xóa tất cả các nút trùng giá trị trong danh sách liên kết (chỉ giữ lại một nút duy nhất với mỗi giá trị riêng biệt, ví dụ danh sách bao gồm 1, 1, 2, 3, 2 sẽ bị xóa thành 1, 2, 3).
void removeDuplicateElements(**List*** &L);
 10. In tất cả các phần tử của danh sách liên kết.
void printElement(**List*** L);
- Các bạn cài đặt lại các hàm trên với danh sách liên kết đôi được định nghĩa như sau (hình 2)

```

struct dNode{
    int key;
    dNode* p_next;
    dNode* p_prev;
};

struct DoubleLinkedList{
    dNode* p_head;
    dNode* p_tail;
};

```

Hình 2: Danh sách liên kết đôi

2.2 Bài tập vận dụng

Các bạn suy nghĩ thuật toán và giải quyết các bài toán dưới đây. Với bài tập (*), đây là các bài tập lấy điểm cộng, sẽ không bắt buộc các bạn phải thực hiện.

Bài tập 1. Các bạn viết hàm in phần tử chính giữa của danh sách liên kết đơn, trong trường hợp có 2 phần tử chỉ in phần tử thứ hai (sẽ rất tối ưu nếu các bạn chỉ dùng một vòng lặp cho bài toán này).

Ví dụ danh sách liên kết $L = 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ sẽ có phần tử chính giữa là 3, danh sách liên kết $L = 1 \rightarrow 2$ sẽ có phần tử chính giữa là 2.

Bài tập 2. Các bạn viết hàm đảo ngược danh sách liên kết với danh sách liên kết đơn. Lưu ý không được tạo thêm danh sách mới và không được thay đổi giá trị của các nút trong danh sách.

Ví dụ danh sách liên kết $L = 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ sẽ có danh sách đảo ngược là $L = 5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1$.

Bài tập 3 (*). Cho một danh sách liên kết đơn và hai vị trí left, right trong đó $0 \leq \text{left} \leq \text{right} \leq (n - 1)$, với n là độ dài của danh sách liên kết. Các bạn viết hàm đảo ngược danh sách liên kết từ vị trí left đến vị trí right.

Ví dụ danh sách liên kết $L = 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$, left = 1, right = 3 sẽ có danh sách đảo ngược là $L = 1 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 5$.

Các bài tập 2 và 3 có thể được cài đặt bằng cách sử dụng đệ quy, các bạn thử suy nghĩ và luyện tập cài đặt các bài này bằng thuật toán đệ quy.

Hết