

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
BỘ MÔN KHOA HỌC MÁY TÍNH

Mã đề
DH 20133 - 02



Họ tên: Lớp: SHSV:	ĐỀ THI MÔN: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT Ngày thi:/...../..... Thời gian 90' (Sinh viên được sử dụng tài liệu)	Hà nội, / / <i>Trưởng bộ môn</i>
---	---	---

Bài 1.

a) Cho **biểu thức trung tố** sau

$$3 + a / (2 - c + b) - a$$

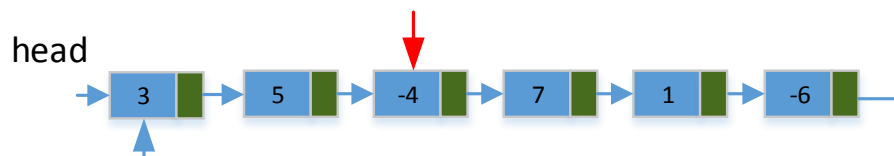
Hãy xây **dựng cây biểu thức** tương ứng

- b) Giả sử bạn có một câu nhị phân chỉ gồm có các **nút trong có 2 con hoặc nút lá**. Khi **duyet cây theo thứ tự** trước từ gốc ta sẽ ký hiệu là L nếu đi theo con trái và R nếu đi theo con phải.
Với thứ tự **duyet trước** thu được của một cây là **LLRRRLLRR** thì **chiều cao** của cây này là?

Bài 2. Danh sách liên kết đơn chứa n số nguyên

```
struct Node
{
    int data;
    struct Node *pNext;
}
```

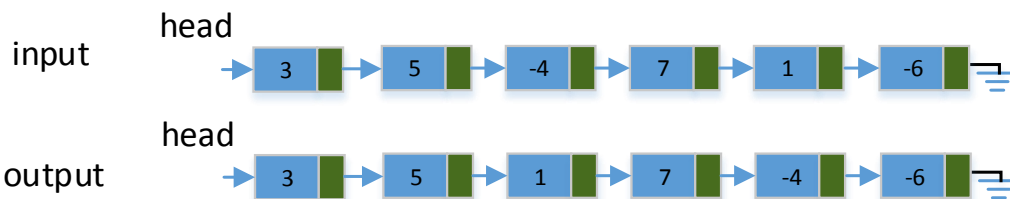
- a) Giả sử danh sách liên kết đơn này được nối vòng. Hãy viết hàm tìm và trả về phần tử giữa danh sách một cách nhanh nhất



```
struct Node findMid(struct Node *head)
```

- b) Hoàn thiện hàm sắp xếp lại danh sách sao cho các số âm được đẩy về cuối dãy và các số dương ở đầu dãy. Hàm cần có thời gian thực hiện cỡ $O(n)$ và không được sử dụng thêm bộ nhớ phụ (ngoại trừ một vài biến và con trỏ phụ)

```
void reorganize(struct Node **head)
```



Bài 3. STACK QUEUE

- Khi biểu diễn ngăn xếp trong trường hợp tổng quát ta nên dùng Mảng hay danh sách liên kết? Vì sao?
- Cho tập hợp gồm n điểm đen và n điểm trắng cách đều nhau. Hãy mô tả thuật toán kết nối một điểm đen với một điểm trắng sao cho tổng khoảng cách là nhỏ nhất.



- Cho một danh sách gồm thời gian của n buổi họp trong một ngày. Hãy xây dựng hàm tìm số phòng cần sử dụng **ít nhất** để đảm bảo các cuộc họp trùng nhau về thời gian sẽ được bố trí ở các phòng khác nhau.

```
struct Meeting
{
    int start, finish;
}

int findNumberOfRooms(struct Meeting A[], int n)
```

Để cho đơn giản, ta tính thời gian bắt đầu và kết thúc (start, finish) bằng số phút chênh lệch so với 00:00 của ngày hôm đó.

- Bài 4.** Cho một mảng chứa n số nguyên bất kỳ, hãy viết hàm tìm và in ra màn hình **dãy con giảm lớn nhất**.
Đánh giá theo O lớn về thời gian thực hiện và bộ nhớ sử dụng trong trường hợp tồi nhất.

```
void maxDecrementalSequence(int A[], int n)
```

- Bài 5.** Hãy xây dựng hàm tính x^n với ($n \geq 0, x > 0$) với thời gian cỡ $O(\log n)$

- Bài 6.** Cây nhị phân

```
struct BNode
{
    int data;
    struct BNode *left, *right, *parent;
}
```

- Viết hàm in ra các nút lá trên cây mà **không dùng đệ quy**

```
void printLeaves(struct BNode *root)
```

- Viết hàm đếm số nút trong có đầy đủ 2 con trên cây nhị phân

```
int countNodes(struct BNODE *root)
```



TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
BỘ MÔN KHOA HỌC MÁY TÍNH

Họ tên: Lớp: SHSV:	ĐỀ THI MÔN: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT Ngày thi:/...../.... Thời gian 90' (Sinh viên được sử dụng tài liệu)	Hà nội, / / <i>Trưởng bộ môn</i>
---	--	---

Bài 1.

a) Đánh giá độ phức tạp của hàm đệ quy sau theo O-lớn

```
void fn(int n)
{
    if (n <= 0) return 1;
    return fn(n - 1) + fn(n - 1);
}
```

b) Để lưu trữ một danh sách các phần tử, dùng danh sách liên kết hoặc mảng sẽ có ưu và nhược điểm gì? Lấy ví dụ.

c) Cho **biểu thức trung tố** sau (2 đ)

$$2 * a / (b - c) * b + 2 / b$$

Hãy xây **tim biểu thức dạng hậu tố** và **dựng cây biểu thức** tương ứng

d) Trong một văn bản HTML các **tag là hợp lệ nếu có đủ thẻ mở và thẻ đóng**.

VD <head></head>

Hãy mô tả thuật toán dùng để **kiểm tra văn bản HTML có hợp lệ hay không**, và nếu không hợp lệ thì tag nào là tag không hợp lệ đầu tiên (tag mở thiếu tag đóng hoặc tag đóng thiếu tag mở)

Bài 2. Phần tử **trung vị** (median) là phần tử có giá trị không nhỏ hơn, cũng không lớn hơn các phần tử còn lại. Ví dụ:

- cho danh sách 4 phần tử sau: 3,7,2,9 thì phần tử trung vị là 3
- cho danh sách 5 phần tử sau: 3,5,7,2,9 thì phần tử trung vị là 5

Giả sử ta cần cái đặt STACK để thực hiện các thao tác

- **push:** đẩy 1 phần tử vào STACK
- **pop:** Lấy ra 1 phần tử khỏi STACK
- **getMedian:** Trả về giá trị trung vị của các phần tử
- **size:** Trả về số lượng phần tử trong STACK

Hãy mô tả cấu trúc dữ liệu cũng như cách thực hiện các thao tác này sao cho **thời gian thực hiện của các thao tác không quá $O(1)$**

Bài 3.

- a) Vẽ cây AVL thu được sau khi thực hiện việc **thêm liên tiếp** các phần tử sau vào cây AVL rỗng
31, 19, 24, 12, 17, 25, 35, 37, 14
- b) Khi thêm lần lượt n phần tử vào cây AVL rỗng thì **số lần phải thực hiện điều chỉnh cây tối thiểu và tối đa** là bao nhiêu lần?
- c) Cho khai báo một nút trên cây AVL như sau, hãy hoàn thiện hàm **tìm và trả về 1 nút lá ở mức sâu nhất** trên cây

```
struct AVLNode
{
    int data;
    int balanceinfo; //0 balanced, -1 left , +1 right
    struct AVLNode * left, *right, *parent;
};

struct AVLNode deepestLeaf(struct AVLNode * root)
```

Bài 4. Cho một xâu ký tự **chỉ chứa các ký tự là chữ cái thường** từ a -> z. Hãy mô tả hàm thực hiện việc **xáo trộn xâu ký tự** này sao cho các **ký tự kề nhau không được giống nhau**.

Trong trường hợp nào ta không thể thực hiện xáo trộn đảm bảo yêu cầu trên ?

VD. Xâu abbcda thì một xáo trộn có thể sẽ là abcadb

Bài 5. Trong một công ty mỗi nhân viên sẽ có một **quản lý trực tiếp**, được biểu diễn bằng cặp **<nhân viên>, <quản lý>**

Một quản lý có thể là nhân viên dưới quyền quản lý của một nhân viên bậc cao hơn. Mỗi một nhân viên phải báo cáo với người quản lý của mình (trực tiếp hoặc gián tiếp).

Giả sử bài toán cần **liệt kê các nhân viên không thuộc quyền quản lý của nhân viên A**. Hãy mô tả **cấu trúc dữ liệu và thuật toán phù hợp** để lưu trữ danh sách nhân viên và thực hiện truy vấn tìm nhân viên không thuộc quyền quản lý của một nhân viên cho trước một cách nhanh nhất.

VD. (A,B), (B,C), (D,C), (E,C)

Nhân viên không thuộc quyền quản lý của B là C, D, E

Nhân viên không thuộc quyền quản lý của A là B, C, D, E

Nhân viên không thuộc quyền quản lý của C là rỗng vì C quản lý tất cả các nhân viên khác

Bài 6. Hoàn thiện hàm **trộn 2 danh sách liên kết đơn đã sắp xếp thành danh sách sắp xếp**

```
struct Node
{
    int data;
    struct Node * pNext;
};

struct Node * mergeLists(struct Node * List1, struct Node * List2)
```



TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
BỘ MÔN KHOA HỌC MÁY TÍNH

Họ tên: Lớp: SHSV:	ĐỀ THI MÔN: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT Ngày thi:/...../..... Thời gian 90' (Sinh viên được sử dụng tài liệu)	Hà nội, / / Trưởng bộ môn
---	---	---

Bài 1.

- a) Trình bày sự khác biệt giữa mảng cấp phát bộ nhớ động và mảng cấp phát tĩnh? Khi nào dùng mảng cấp phát động, mảng cấp phát tĩnh, cho ví dụ.
- b) Đánh giá độ phức tạp của hàm đệ quy sau theo O-lớn

```
void fx(int n)
{
    if (n <= 0) return 1;
    return fx(n/2) + fx(n/2) + x;
}
```

- c) Để lưu trữ một danh sách các phần tử, dùng danh sách liên kết hoặc mảng sẽ có ưu và nhược điểm gì? Lấy ví dụ.
- d) Cho **biểu thức trung tố** sau (2 đ)

$$2 * a/b - c * b + 2/b$$

Hãy xây **tìm biểu thức dạng hậu tố** và **dựng cây biểu thức** tương ứng

- e) Trong một văn bản HTML các **tag** là **hợp lệ** nếu có đủ **thẻ mở** và **thẻ đóng**.

VD <head></head>

Hãy mô tả thuật toán dùng để **kiểm tra văn bản HTML có hợp lệ hay không**, và nếu không hợp lệ thì tag nào là tag không hợp lệ đầu tiên (tag mở thiếu tag đóng hoặc tag đóng thiếu tag mở)

Bài 2.

- a) Vẽ cây thu được sau khi thực hiện việc **thêm liên tiếp** các phần tử sau vào cây nhị phân tìm kiếm rỗng

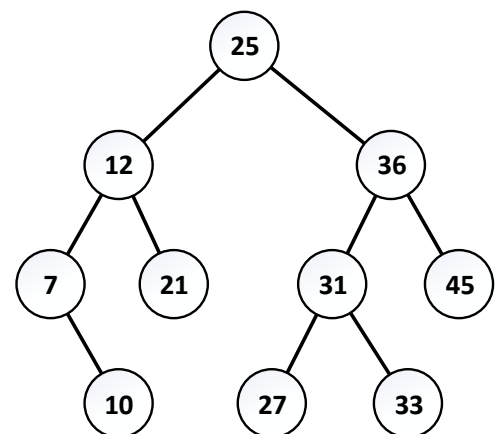
31, 19, 24, 12, 17, 25, 35, 37, 14

- b) Cho cây nhị phân tìm kiếm như hình bên, hãy vẽ cây thu được sau **khi xóa liên tiếp 25 và 36**

- c) Khi thêm lần lượt n phần tử vào cây nhị phân tìm kiếm rỗng thì **chiều cao tối thiểu và tối đa** là bao nhiêu?

- d) Viết hàm đếm số lượng nút **có đầy đủ hai con** trên cây nhị phân

Bài 3. Cho một danh sách số nguyên mà mỗi cặp phần tử liên tiếp nhau có giá trị chênh lệch là -1 hoặc +1, và một số nguyên x. Hãy đưa ra thuật toán tìm vị trí của x trong danh sách một cách nhanh nhất mà không cần phải duyệt toàn bộ các phần tử.
VD. Danh sách 1,2,1,2,3,4,3,2,1,2,3,4,5,4,3,4 và x=5 thuật toán trả về 12 (bắt đầu từ 0)



Bài 4. Cho tập hợp gồm n điểm đen và n điểm trắng cách đều nhau. Hãy mô tả thuật toán kết nối một điểm đen với một điểm trắng sao cho tổng khoảng cách là nhỏ nhất.



Bài 5. Hoàn thiện hàm trộn 2 danh sách liên kết đơn đã sắp xếp thành danh sách sắp xếp

```
struct Node
{
    int data;
    struct Node * pNext;
};

struct Node * mergeLists(struct Node * List1, struct Node * List2)
```

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
BỘ MÔN KHOA HỌC MÁY TÍNH

Mã đề
DH 20133 - 01



Họ tên:	ĐỀ THI MÔN: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT	Hà nội, / /
Lớp:	Ngày thi:/...../.....	Trường bộ môn
SHSV:	Thời gian 90' (Sinh viên được sử dụng tài liệu)	

Bài 1.

- a) Cho **biểu thức trung tố** sau

$$(3 + a)/2 - c + 2/a$$

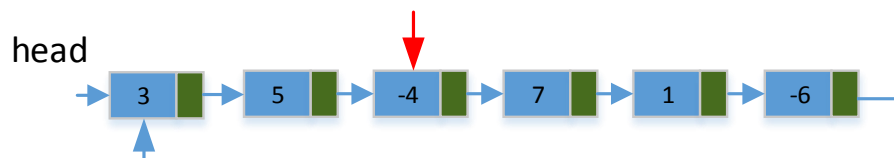
Hãy xây **dựng cây biểu thức** tương ứng

- b) Giả sử bạn có một câu nhị phân chỉ gồm có các **nút trong có 2 con hoặc nút lá**. Khi **duyet cây theo thứ tự** trước từ gốc ta sẽ ký hiệu là L nếu đi theo con trái và R nếu đi theo con phải.
Với thứ tự **duyet trước** thu được của một cây là **LLRLRLRLR** thì **chiều cao** của cây này là?

Bài 2. Danh sách liên kết đơn chứa n số nguyên

```
struct Node
{
    int data;
    struct Node *pNext;
}
```

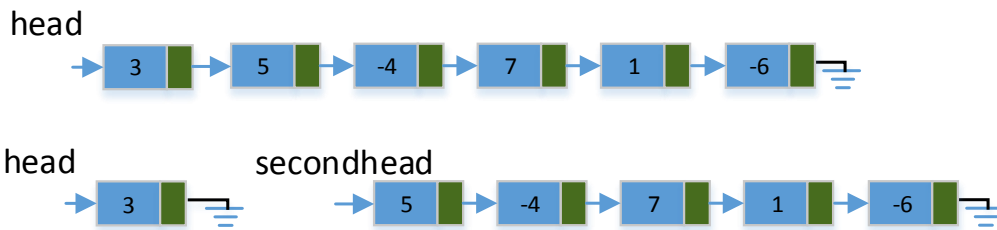
- a) Giả sử danh sách liên kết đơn này được nối vòng. Hãy viết hàm tìm và trả về phần tử giữa danh sách một cách nhanh nhất



```
struct Node findMid(struct Node *head)
```

- b) Hoàn thiện hàm **tìm vị trí cắt danh sách thành 2 nửa** sao cho tổng độ chênh lệch giữa 2 nửa là nhỏ nhất

```
void splitLinkedList(struct Node **head, struct Node **secondhead)
```



Bài 3. STACK QUEUE

- Khi biểu diễn ngăn xếp trong trường hợp tổng quát ta nên dùng Mảng hay danh sách liên kết? Vì sao?
- Cho tập hợp gồm n điểm đen và n điểm trắng cách đều nhau. Hãy mô tả thuật toán kết nối một điểm đen với một điểm trắng sao cho tổng khoảng cách là nhỏ nhất.



- Cho một danh sách gồm thời gian của n buổi họp trong một ngày. Hãy xây dựng hàm tìm số phòng cần sử dụng **ít nhất** để đảm bảo các cuộc họp trùng nhau về thời gian sẽ được bố trí ở các phòng khác nhau.

```

struct Meeting
{
    int start, finish;
}

int findNumberOfRooms(struct Meeting A[], int n)

```

Để cho đơn giản, ta tính thời gian bắt đầu và kết thúc (start, finish) bằng số phút chênh lệch so với 00:00 của ngày hôm đó.

- Bài 4.** Cho một mảng chứa n số nguyên bất kỳ, hãy viết hàm tìm và in ra màn hình **dãy con tăng lớn nhất**. Đánh giá theo O lớn về thời gian thực hiện và bộ nhớ sử dụng trong trường hợp tồi nhất.

```

void maxIncrementalSequence(int A[], int n)

```

- Bài 5.** Hãy xây dựng hàm tính x^n với ($n \geq 0, x > 0$) với thời gian cỡ $O(\log n)$

- Bài 6.** Cây nhị phân

```

struct BNode
{
    int data;
    struct BNode *left, *right, *parent;
}

```

- Viết hàm in ra các nút lá trên cây mà **không dùng đệ quy**

```

void printLeaves(struct BNode *root)

```

- Viết hàm đếm số nút trong có đầy đủ 2 con trên cây nhị phân

```

int countNodes(struct BNode *root)

```