

Bài tập phần II

Phần: Các cấu trúc dữ liệu cơ bản- Mảng, Danh sách liên kết, danh sách tuyến tính

Bài 1. Kiểu dữ liệu là gì ? cho ví dụ minh họa.

Bài 2. Kiểu dữ liệu trừu tượng là gì? Nó khác gì so với định nghĩa kiểu dữ liệu?

Bài 3. Cấu trúc dữ liệu là gì ?

Bài 4. Phân loại cấu trúc dữ liệu? So sánh đặc điểm của các phân loại đó.

Bài 5. Trình bày ưu nhược điểm của mảng (bao gồm cả mảng cấp phát tĩnh và mảng cấp phát động).

Bài 6. Viết hàm để thực hiện thêm phần tử vào vị trí sau phần tử thứ k trong hai trường hợp : mảng, và danh sách liên kết đơn.

Bài 7. Viết hàm **reverse** để in ra các phần tử trong danh sách liên kết đơn theo thứ tự đảo ngược

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

//pHead là con trỏ đến đầu danh sách cần đảo ngược
void reverse(NODE *pHead)
{
    //Thân hàm
}
```

Ví dụ: với danh sách ban đầu chứa các phần tử {2, 27, 5, 14, 9}, thì kết quả sẽ là 9, 14, 5, 27, 2

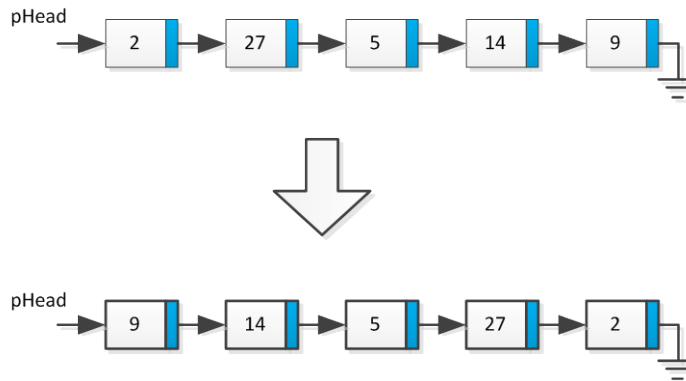
Hãy đánh giá thời gian thực hiện thuật toán của bạn theo O-lớn.

Bài 8. Hoàn thiện phần thân hàm **reverse** để đảo ngược danh sách liên kết đơn

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

//pHead là con trỏ đến đầu danh sách cần đảo ngược
void reverse(NODE *&pHead)
{
    //Thân hàm
}
```

Ví dụ với danh sách ban đầu chứa các phần tử {2, 27, 5, 14, 9}, thì kết quả sẽ là danh sách {9, 14, 5, 27, 2} ở dưới



Hãy đánh giá thời gian thực hiện thuật toán của bạn theo O-lớn.

Bài 9. Viết lại các hàm thực hiện các thao tác chèn, tìm kiếm và xóa phần tử trên danh sách liên kết đơn dùng vòng lặp thay vì dùng đệ quy.

Bài 10. Viết lại hàm xóa phần tử trong danh sách liên kết đơn mà không cần dùng thêm các hàm `search_list`, `predecessor_list`.

Bài 11. Cho một dãy số nguyên bất kỳ có số lượng phần tử lớn hơn 2. Hãy viết hàm **find** để tìm và in ra màn hình hai cặp phần tử có độ chênh lệch lớn nhất và nhỏ nhất trong dãy đã cho.

Độ chênh lệch giữa 2 số a, b được định nghĩa là $d = |a - b|$

```
//A là tên mảng chứa các phần tử
//n là số lượng phần tử trong mảng, n>2
void find(int A[], int n)
{
    //thân hàm
}
```

Ví dụ với dãy {1,5,3,2,7,4} thì cặp phần tử có chênh lệch lớn nhất là (1, 7), cặp phần tử có độ chênh lệch nhỏ nhất là (2,3). Trong trường hợp có nhiều cặp chỉ cần đưa ra 1 cặp bất kỳ thỏa mãn.

Hãy đưa ra đánh giá thời gian thực hiện của thuật toán của bạn theo O-lớn

Bài 12. Cho một dãy các số nguyên khác nhau đã được sắp theo thứ tự tăng dần. Viết chương trình kiểm tra xem có tồn tại phần tử nào mà giá trị của nó đúng bằng vị trí của nó.

Ví dụ. trong dãy {-10,-3, 3, 5, 7}, ta tìm được $a_3 = 3$.

Trong dãy {2, 3, 4, 5, 6, 7}, thì không tồn tại phần tử nào như vậy

Chú ý: Vị trí của phần tử được tính bắt đầu từ 1

Bài 13. Để biểu diễn kiểu dữ liệu ADT chuỗi ký tự trong máy tính người ta có thể làm theo những cách sau

1. Lưu trữ bằng mảng, dùng một ký tự đặc biệt để báo hiệu kết thúc chuỗi (ký tự '\0'). Mảng phải được khai báo có số phần tử tối đa là lớn hơn bằng số lượng ký tự của chuỗi.
2. Lưu trữ bằng mảng, phần tử đầu tiên trong mảng ta dùng để chứa số ký tự hiện thời của chuỗi ký tự, còn các phần tử tiếp theo sẽ là các ký tự.

3. Dùng danh sách liên kết để chứa các ký tự, kết thúc xâu khi phần tử cuối cùng là NULL.

Hãy nhận xét về hiệu quả của mỗi phương pháp theo các tiêu chí

- a) Kích thước bộ nhớ cần dùng để lưu trữ xâu ký tự
- b) Các ký tự mà xâu có thể biểu diễn
- c) Thời gian để truy cập vào ký tự thứ i trong xâu
- d) Thời gian thực hiện phép chèn, xóa ký tự trong xâu

Bài 14. Giả sử chúng ta có con trỏ trỏ đến phần tử cần xóa trong danh sách, có cách nào khác để xóa phần tử đó khỏi danh sách mà không cần phải duyệt danh sách để tìm phần tử đứng trước nó không? Nếu có hãy mô tả phương pháp của bạn.

Bài 15. Viết chương trình tìm phần tử có giá trị lớn nhất, nhỏ nhất trong danh sách móc nối đơn.

Bài 16. Xây dựng chương trình biểu diễn đa thức $P_n(x) = a_nx^n + a_{n-1}x^{n-1} + \dots + a_1x + a_0$ với các thao tác cơ bản như hiển thị, cộng, trừ, nhân hai đa thức.

Bài 17. So sánh ưu nhược điểm của mảng và cấu trúc liên kết khi dùng để lưu trữ kiểu dữ liệu trừu tượng danh sách tuyến tính.

Bài 18. Cần phải lưu trữ một danh sách tuyến tính là thông tin về khách hàng trong các ngày trong tháng ở siêu thị. Ta sẽ chọn cấu trúc dữ liệu nào để lưu trữ sao cho thao tác tìm kiếm là nhanh nhất và tiết kiệm bộ nhớ nhất nếu chúng ta biết:

- a) Không biết trước số lượng khách hàng tối đa
- b) Biết trước số lượng khách hàng tối đa nhưng số lượng khách hàng giữa các ngày khác nhau biến động rất lớn
- c) Biết trước số lượng khách hàng tối đa, và số lượng khách hàng trong các ngày chênh lệch nhau không nhiều

Các cấu trúc có thể được lựa chọn là: Mảng tĩnh, mảng động, cấu trúc móc nối.

Hãy giải thích lựa chọn của bạn trong từng trường hợp.

Bài 19. Cài đặt chương trình mô phỏng bài toán *Josephus* trong slide.

Bài 20. Cho một dãy số gồm n số, hãy viết chương trình in ra các phần tử trong dãy theo chiều tăng dần tần số xuất hiện

Ví dụ: dãy 1, 3, 4, 5, 7, 2, 3, 5 thì ta sẽ in ra 1, 2, 4, 7, 3, 5

Bài 21. Cài đặt các hàm thực hiện thao tác thêm, xóa và tìm kiếm đối với danh sách liên kết đôi.

Bài 22. Cài đặt các hàm thực hiện thao tác thêm, xóa và tìm kiếm đối với danh sách liên kết đôi nối vòng (danh sách nối đôi sử dụng nút đầu giả).

Bài 23. Viết chương trình cài đặt mảng động để lưu trữ danh sách các phần tử theo cách sau:

- Ban đầu ta cấp phát bộ nhớ động là 10 phần tử
- Nếu mà ta phải thêm một phần tử mới khi mảng đã đầy thì ta gấp đôi kích thước mảng và copy toàn bộ các phần tử của mảng cũ vào nửa đầu mảng mới, sau đó thêm phần tử mới vào mảng mới.
- Nếu mà xóa một phần tử mà sau khi xóa thì số phần tử của mảng nhỏ hơn $1/2$ kích thước của mảng thì ta tiến hành tạo mảng mới với kích thước bằng $\frac{1}{2}$ kích thước mảng cũ. Sau đó copy toàn bộ các phần tử mảng cũ sang mảng mới.

Hãy đánh giá thời gian thực hiện trong trường hợp tốt nhất, tồi nhất cả các thao tác thêm và xóa phần tử.

Chỉ ra một trường hợp mà cách làm như vậy cho kết quả rất tồi. Bạn có cách nào để cải thiện hiệu quả ?

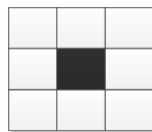
Bài 24. Cho hai chuỗi ký tự s_1 và s_2 . Hãy viết hàm **stringmatch** để kiểm tra xem chuỗi ký tự s_2 có xuất hiện trong s_1 hay không. Nếu có thì trả về vị trí xuất hiện đầu tiên của nó, ngược lại thì trả về giá trị là -1.

Ví dụ: $s_1 = \text{"AbbAbbbabbb"}$, $s_2 = \text{"ab"}$ thì hàm **stringmatch**(s_1, s_2) sẽ trả về giá trị 7

Hãy đánh giá hiệu quả của thuật toán của bạn.

Bài 25. Trò chơi dò mìn

Giả sử chúng ta mô tả bãi mìn bằng một ma trận với kích thước $N \times M$ (N hàng và M cột). Những vị trí có mìn được đánh dấu bằng ký tự '*', còn những vị trí không có mìn là ký tự '.'. Hãy xây dựng một ma trận tương đương để mô tả các cảnh báo có mìn của các ô xung quanh. Với mỗi ô có mìn thì ta vẫn biểu diễn bằng ký tự '*', còn những ô xung quanh ta biểu diễn bằng các chữ số mô tả số ô có mìn mà lân cận với nó.




Ở đây 1 ô sẽ lân cận với 8 ô xung quanh

Ví dụ

Bãi mìn kích thước 4×5 được biểu diễn lại như sau

.	.	.	*	.
*	.	*	.	*
.	*	.	.	.
.	.	.	.	*



1	2	2	*	2
*	3	*	3	*
2	*	2	3	2
1	1	1	1	*

Hãy xây dựng hàm

Calculate(char MineField[10][10], char Output[10][10], int N, int M)

Để tính toán các giá trị cho mảng Output với đầu vào là mảng mô tả bãi mìn MineField

$1 \leq N, M \leq 10$ là kích thước thực sự của bãi mìn hiện tại

Bài 26. Cho kiểu dữ liệu trừu tượng tập hợp với

- Các phần tử của tập hợp là các số nguyên
- Các phép toán của tập hợp gồm :

- **Member**(x,S): Kiểm tra xem phần tử x có thuộc tập hợp S hay không
- **Union**(A,B): Hợp của hai tập hợp A, B, trả về một tập hợp $A \cup B$ thông qua tập hợp A, hoặc B, hoặc hàm
- **Intersection**(A,B): Giao của hai tập hợp, trả về $A \cap B$ thông qua tập hợp A, hoặc B, hoặc hàm
- **Substract**(A,B): Hiệu của hai tập hợp, trả về $A - B$ thông qua tập hợp A, hoặc B, hoặc hàm
- **Insert**(x,S): thêm phần tử x vào tập hợp S
- **Delete**(x,S) : loại bỏ phần tử x khỏi tập S (nếu x có trong S)

Hãy cài đặt ADT trên sử dụng các phương pháp biểu diễn

- Mảng
- Danh sách liên kết
- Xâu bit (bit 1 biểu diễn phần tử có trong tập hợp và bit 0 biểu diễn phần tử không có)

Chú ý: Đối với cách biểu diễn thứ 3 các phần tử phải được biểu diễn theo một thứ tự xác định.

Các phần tử trong tập hợp không trùng nhau.

Bài 27. Cho một dãy số nguyên dương gồm n số, khoảng cách giữa hai vị trí chính là độ chênh lệch về giá trị của hai số tại vị trí đó.

Ví dụ cho dãy gồm 5 số 1, 3, 4, 3, 7. Chênh lệch giữa số thứ 1 và 4 là $3 - 1 = 2$.

Chọn một vị trí k bất trong dãy n số, tổng độ chênh lệch của vị trí k với tất cả $n - 1$ vị trí còn lại được tính bằng tổng độ chênh lệch của số ở vị trí k tới $n - 1$ vị trí còn lại.

Ví dụ với k=2 (số giá trị 3) thì tổng độ chênh lệch ứng với 2 sẽ là $2+1+0+4=7$

Hãy viết chương trình tìm vị trí k sao cho có tổng độ chênh lệch là nhỏ nhất.

Bài 28. Viết chương trình đảo ngược thứ tự các từ trong một câu được nhập vào từ bàn phím.

Ví dụ. Câu đầu vào là “the quick brown fox jumps over the lazy dog”

Thì kết quả hiển thị ra màn hình sẽ là câu “dog lazy the over jumps fox brown quick the”

Bài 29. Nhập vào từ bàn phím một dãy số thực gồm n số ($0 < n < 100$). Thực hiện các công việc sau:

- In ra màn hình dãy số vừa nhập.
- Tìm và in ra màn hình các số âm trong dãy, nếu không có số nào thì in ra thông báo “Dãy không có số âm”.
- Tìm và in ra màn hình giá trị phần tử dương lớn nhất, nhỏ nhất trong dãy.
- Nhập vào một giá trị k, xóa tất cả các phần tử có giá trị lớn hơn k trong dãy. In ra màn hình dãy sau khi xóa.

Bài 30. Giáo viên chủ nhiệm muốn quản lý thông tin về tình hình học tập của tất cả các thành viên trong lớp. Thông tin về mỗi thành viên gồm:

- Họ tên
- Giới tính
- Ngày sinh (lưu bằng chuỗi ký tự)

- Địa chỉ
- Điểm trung bình cả năm

Hãy xây dựng một cấu trúc dữ liệu phù hợp để lưu thông tin về các thành viên của lớp. Sử dụng cấu trúc do bạn tự định nghĩa để nhập vào 10 thành viên của lớp. Sau đó thực hiện các công việc sau:

- In ra thông tin về 10 thành viên vừa nhập
- Tìm và in ra thông tin về những thành viên có điểm trung bình cả năm nhỏ hơn 4.0

In ra thông tin về thành viên có điểm trung bình cả năm cao nhất của lớp.

Bài 31. Một cửa hàng cho thuê băng đĩa muốn quản lý thông tin về các băng đĩa mà cửa hàng đó cho thuê (giả sử với mỗi phim thì cửa hàng chỉ có một bộ đĩa duy nhất). Mỗi một đĩa có các thông tin sau:

- Mã số : là một xâu ký tự, mỗi đĩa được biểu diễn bởi một xâu duy nhất
- Tiêu đề : thông tin về tên đĩa
- Loại đĩa: là loại CD hay VCD hay DVD (gợi ý: sử dụng số nguyên để biểu diễn 1-CD, 2-VCD, 3-DVD)

Do số lượng đĩa cho thuê trong từng ngày là biến đổi rất nhiều, nên các tối ưu là lưu các đĩa cho thuê bằng cấu trúc danh sách móc nối. Bạn hãy định nghĩa cấu trúc danh sách móc nối để lưu thông tin về các đĩa cho thuê. Thêm vào danh sách các thông tin về các đĩa được thuê, ví dụ

1. "D001", "Kungfu panda", "VCD"
2. "D003", "WALL-E", "DVD"
3. "D005", "Earth", "VCD"

Bây giờ khách hàng thuê đĩa có mã "D003" trả, hãy cập nhật lại danh sách cho thuê, xóa nút có thông tin về đĩa D003 đó đi (gợi ý: thực hiện tìm kiếm trên danh sách móc nối để tìm D003). In ra màn hình các đĩa đã được thuê trong ngày.

Bài 32. Cho các con trỏ sau

```
int *p, *q, a,b;
float *f,x;
a=5;
b=7;
x=9;
```

Chuyện gì sẽ xảy ra khi ta thực hiện các câu lệnh sau, giải thích

1. `p=&a;`
 `*p=45;`
2. `p=&a;` `q=&b;`
 `q=p;`

```
*q=5;
```

```
3. x=a;  
   f=&x;  
   *f=4.5;
```

```
4. f=&a;  
   p=f;
```

Bài 33. Cho các dữ liệu ban đầu như sau

```
int a[] = { 3, 8, -4, 6, 2, 6 };  
int *p1 = a;  
int *p3 = &a[5];  
int *p2 = a+2;
```

Thực hiện tuần tự các lệnh sau, hãy cho biết nội dung của mảng a sau khi thực hiện mỗi lệnh

```
1. *p1=5;  
2. *p2=7;  
3. *p3 = *p1+*p2;  
4. p1=p2-1;  
5. *p1=55;  
6. *(p1+3)=99;
```