



Cau truc du lieu va giai thuat_ Nhóm 08

[Nhà của tôi](#) / [Các khoá học của tôi](#) / [DASA230179_21_1_08](#) / [Kiểm tra quá trình](#) / [Nhóm câu hỏi 2](#)

Bắt đầu vào lúc	Saturday, 11 December 2021, 11:22 AM
Trạng thái	Đã xong
Kết thúc lúc	Saturday, 11 December 2021, 4:39 PM
Thời gian thực hiện	5 giờ 17 phút

Câu hỏi 1

Hoàn thành

Đạt điểm 1,00

Hiện thực hàm xác định giá trị Key có tồn tại trong danh sách liên kết đơn head. Nếu tìm có giá trị key thì xuất thứ tự key trong danh sách (nếu có nhiều vị trí thì xuất vị trí đầu tiên), nếu không có thì xuất -1

head: 1->2->3 ->4 ->5 ->6 ->7

Key =4

Output: 3

```
#include <stdio.h>

struct Node {
    int data;
    Node* next;
};

int searchInList(Node*head, int Key)
{
    //your code here
    int index = -1, dem = -1;
    if (head == nullptr)
    {
        return index;
    }
    Node* temp = head;
    while (temp != nullptr)
    {
        dem++;
        if (temp->data == Key)
        {
            break;
        }
        temp = temp->next;
    }
    if (temp->data == Key)
    {
        index = dem;
    }
    return index;
    //end your code
}
```

Câu hỏi 2

Hoàn thành

Đạt điểm 1,00

Cho 2 danh sách liên kết head1, head2 đã được sắp xếp theo thứ tự không giảm, hãy viết chương trình để gộp 2 danh sách liên kết này lại thành 1 danh sách liên kết theo thứ tự không giảm

```
#include <stdio.h>

struct Node {
    int data;
    Node* next;
};

Node* mergeList(Node*head1, Node*head2)
{
    //your code here
    Node* result = NULL;
    if (head1 == NULL)
        return(head2);
    else if (head2 == NULL)
        return(head1);
    if (head1->data <= head2->data)
    {
        result = head1;
        result->next = mergeList(head1->next, head2);
    }
    else
    {
        result = head2;
        result->next = mergeList(head1, head2->next);
    }
    return(result);
    //end your code
}
```

Câu hỏi 3

Hoàn thành

Đạt điểm 1,00

Độ phức tạp của thuật toán tìm kiếm nhị phân

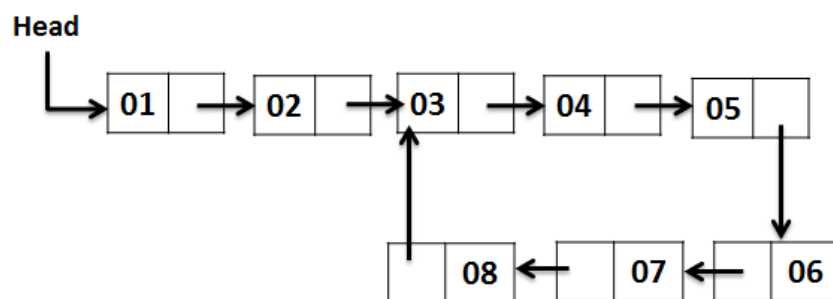
- ☐ a. $O(N*N)$
- ☐ b. $O(N)$
- ☒ c. $O(\log N)$
- ☒ d. $O(1)$

Câu hỏi 4

Hoàn thành

Đạt điểm 1,00

Một bạn hiện thực danh sách liên kết đơn có thể chứa các vòng như hình



Giả sử danh sách chỉ có tối đa 1 vòng. Hãy viết chương trình để xuất ra giá trị bắt đầu và kết thúc của vòng

Ví dụ: input là danh sách liên kết như hình

Output: 3

Nếu không có vòng thì output -1

```

#include <stdio.h>

struct Node {
    int data;
    Node* next;
};

int checkCycle(Node*head)
{
    //your code here
    if (head == NULL || head->next == NULL)
        return -1;

    Node* slow = head, * fast = head;
    slow = slow->next;
    fast = fast->next->next;
    while (fast && fast->next) {

        if (slow == fast)
            break;
        slow = slow->next;
        fast = fast->next->next;
    }

    if (slow != fast)
        return -1;

    slow = head;
    while (slow != fast) {
        slow = slow->next;

```

```
    fast = fast->next;  
}  
return slow->data;  
//end your code  
}
```

Câu hỏi 5

Không trả lời

Đạt điểm 1,00

Viết chương trình để chuẩn hóa đường dẫn trong unix:

`"/ab/./" --> tại thư mục ab`

`"/ab/bc/.." --> di chuyển thư mục cha của thư mục bc --> thư mục ab`

`"/" --> "/"`

Input : /nam/

Output : /nam

Input : /aa/./bb/./../cc/

Output : /cc

Input : /aa/..

Output: /

Input : /aa/./

Output : /

Input : /.././.././.././aa

Output : /aa

Input : /aa/./bb/./cc/./dd/

Output : /aa/bb/cc/dd

Input : /aa/./.././.././../.

Output: /

Input : /aa//bb//cc/////dd

Output : /aa/bb/cc/dd

Hãy sử dụng Queue để hiện thực yêu cầu trên

Câu hỏi 6

Hoàn thành

Đạt điểm 1,00

Kết quả của đoạn chương trình sau

```
1 | int fun(int n)
2 | {
3 |     if (n == 4)
4 |         return n;
5 |     return 2*fun(n+1);
6 | }
7 |
8 | fun(2)
```

Answer: 16

Câu hỏi 7

Hoàn thành

Đạt điểm 1,00

Tìm kiếm nhị phân là nhóm thuật toán sử dụng phương pháp

- ☐ a. Quy hoạch động
- ☒ b. Chia để trị
- ☐ c. Tìm kiếm vét cạn
- ☐ d. Thuật toán tham lam

Câu hỏi 8

Hoàn thành

Đạt điểm 1,00

Cho giá trị thị trường của mã chứng khoán trong N ngày. Hãy tính chỉ số Span của mã chứng khoán này.

Span là tổng số ngày trước ngày hiện tại có giá thấp hơn hoặc bằng giá hiện tại

Ví dụ:

Input: [100,80,60,70,60,75,85]

Output: [1,1,1,2,1,4,6]

Tại index 0: 100 lớn nhất -> 1

Tại index 1: 80 là giá trị nhỏ nhất -> 1

Tại index 2: 60 là giá trị nhỏ nhất -> 1

Tại index 3: có 60 < 70 -> 2

Tại index 4: index 3 có giá trị 70 > 60 -> 1

Tại index 5: có 60, 70, 60 < 75 -> 4

Tại index 6: có 80,60,70,60,75 <85 -> 6

```
#include <stdio.h>
```

```
struct Node {
```

```
    int data;
```

```
    Node* next;
```

```
};
```

```
struct Queue {
```

```
    // your code here
```

```
Node* top = nullptr;
```

```
Node* bottom = nullptr;
```

```
void insert(int value) {
```

```
    if (top == nullptr) {
```

```
        top = new Node;
```

```
        top->data = value;
```

```
        top->next = nullptr;
```

```
        bottom = top;
```

```
        return;
```

```
    }
```

```
Node* temp;
```



```
temp = new Node;

temp->data = value;

temp->next = top;

top = temp;

}

int pop() {

    if (bottom == nullptr) {

        return -1;

    }

    int value = bottom->data;

    Node* temp = bottom;

    Node* last = top;

    if (top == bottom) {

        Node* temp = top;

        top = bottom = nullptr;

        delete temp;

        return value;

    }

    while (last->next != nullptr && last->next != bottom) {

        last = last->next;

    }

    bottom = last;
```

```
        bottom->next = nullptr;

        delete temp;

        return value;

    }

    bool empty() {

        return (bottom == nullptr);

    }
//end your code
};

int* chisoSpan(int[] arr, int N) {
    //your code here
    int* res = new int[N];

    int i, j;

    Queue q;

    for (i = 0; i < N; i++)

    {

        int count = 0;

        for (j = 0; j < i; j++)

            if (arr[j] < arr[i])

                count++;

        q.insert(count + 1);

    }

    for (i = 0; i < N; i++)

        res[i] = q.pop();

    return res;
//end your code
}
```


Câu hỏi 9

Hoàn thành

Đạt điểm 1,00

Hãy áp dụng cấu trúc dữ liệu Queue để giải quyết vấn đề sau:

Bài toán tìm ký tự không lặp lại đầu tiên trong một chuỗi các ký tự. Cho một mảng một chiều có N ký tự thường (không in hoa). Tại vị trí index thứ i trong mảng phải xác định ký tự nào là không lặp lại trong mảng.

Yêu cầu:

- chỉ lặp trên mảng ban đầu 1 lần duy nhất
- Nếu không có ký tự nào thỏa mãn thì xuất giá trị 0
- Output là 1 mảng ký tự

Input: aabcbcd

Output: a0bbc0d

Giải thích:

- Tại vị trí 0, ký tự a lần đầu xuất hiện --> a
- Vị trí 1, ký tự a đã xuất hiện --> a0
- Vị trí 2, ký tự b lần đầu xuất hiện --> a0b
- Vị trí 3, ký tự c lần đầu xuất hiện, tuy nhiên ký tự b vẫn chưa có lặp lại --> a0bb
- Vị trí 4, ký tự b lặp lại, c mới xuất hiện 1 lần --> a0bbc
- Vị trí 5, ký tự c lặp lại --> a0bbc0
- Vị trí 6 ký tự d lần đầu xuất hiện --> a0bbc0d

```
#include <stdio.h>
```

```
struct Node {
```

```
    int data;
```

```
    Node* next;
```

```
};
```

```
struct Queue {
```

```
    // your code here
```

```
Node* top = nullptr;
```

```
Node* bottom = nullptr;
```

```
void insert(int value) {
```

```
    if (top == nullptr) {
```

```
        top = new Node;
```

```
        top->data = value;
```

```
        top->next = nullptr;
```

```
        bottom = top;
```

```
        return;
```

```
    }
```

```
    Node* temp;
```

```
    temp = new Node;
```

```
    temp->data = value;
```

```
    temp->next = top;
```

```
    top = temp;
```

```
}
```

```
int pop() {
```

```

    if (bottom == nullptr) {
        return -1;
    }

    int value = bottom->data;
    Node* temp = bottom;
    Node* last = top;
    if (top == bottom) {
        Node* temp = top;
        top = bottom = nullptr;
        delete temp;
        return value;
    }

    while (last->next != nullptr && last->next != bottom) {
        last = last->next;
    }

    bottom = last;
    bottom->next = nullptr;
    delete temp;
    return value;
}

bool empty() {
    return (bottom == nullptr);
}

int Size()
{
    if (top == nullptr)
    {
        return 0;
    }

    int size = 1;
    Node* temp = top;
    while (temp->next != nullptr)
    {
        size++;
        temp = temp->next;
    }

    return size;
}

//end your code
};

char* firstCharacter(char[] s, int N) {
    //your code here

    Queue q;

    int i, j;

    int charCount[26] = { 0 };

    char* res = new char[N];

```



```
for (i = 0; s[i]; i++) {  
  
    charCount[s[i] - 'a']++;  
    for (j = 0; j < N; j++)  
    {  
        if (charCount[j] == 1)  
        {  
            q.insert(j + 'a');  
            j = N;  
        }  
        if (charCount[j] == 0)  
        {  
            break;  
        }  
    }  
    if (charCount[j-1] > 1 )  
    {  
        q.insert('0');  
    }  
    res[i] = q.pop();  
  
}  
return res;  
//end your code  
}
```

Câu hỏi 10

Hoàn thành

Đạt điểm 1,00

Thuật toán tìm kiếm nhị phân chỉ có thể áp dụng cho mảng dữ liệu đã sắp xếp theo thứ tự

Hãy chọn một:

- ☐ Đúng
- ☒ Sai

◀ Nhóm câu hỏi 1

Chuyển tới...

Nhóm câu hỏi 3 ▶

