

Midterm exam

Opens: Thứ bảy, 16 Tháng mười 2021, 1:00 PM

Closes: Thứ bảy, 16 Tháng mười 2021, 2:05 PM

Thời gian thi là **60'**.

Hình thức thi: làm trực tiếp trên máy

Ghi chú: Sinh viên phải ghi toàn bộ màn hình của mình trong quá trình làm việc và nộp về để làm mình chứng. Sinh viên có thể sử dụng google meet để ghi hình sử dụng tính năng trình bày toàn màn hình.

Số lần làm bài cho phép: 1

Giới hạn thời gian: 1 giờ

Điều hướng bài kiểm tra

PHẦN TRẢ LỜI NGẮN

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21						

PHẦN TỰ LUẬN (ĐỒNG
THỜI TÍNH ĐIỂM HARMONY
VỚI BÀI TẬP LỚN SỐ 1)

22

[Hoàn thành bài làm ...](#)

Thời gian còn lại 0:59:56

Câu hỏi 1

Chưa được trả lời

Chấm điểm của 1,00

Cờ câu hỏi

Cho các phát biểu sau:

1. Hiện thực danh sách bằng mảng nên chỉ áp dụng cho các danh sách có ít phần tử.
2. Hiện thực danh sách bằng mảng nên chỉ áp dụng cho các danh sách thường xuyên truy cập các phần tử.
3. Hiện thực danh sách bằng mảng phù hợp cho các danh sách thường xuyên diễn ra thao tác thêm và xóa vì tính đơn giản trong hình thức lưu trữ.
4. Việc dịch chuyển một số lượng phần tử không bị tác động nhiều khi ta thêm phần tử vào cuối danh sách.

Số phát biểu đúng là

Chọn một:

- ☐ a. 3
- ☐ b. 0
- ☐ c. 2
- ☐ d. 1

Thời gian còn lại 0:58:58

Câu hỏi 2

Chưa được trả lời

Chấm điểm của 1,00

Cờ câu hỏi

Hãy kéo thả các câu lệnh dưới đây để hoàn thành hàm foo để hàm foo là hàm trả về phần tử lớn nhất trong một mảng các số nguyên **arr** với số lượng phần tử **n**:

```
int foo(int* arr, int n) {
```

```
    if ( ) return arr[0];
```

```
    int maxEle = foo(arr, );
```

```
    return (maxEle > ) ? : arr[n - 1];
```

```
}
```

arr[0]

n == 0

n - 1

maxEle

maxEle + arr[0]

n - 2

maxEle - 1

maxEle + arr[n - 1]

arr[n - 1]

n == 1

Thời gian còn lại 0:56:43

Câu hỏi 3

Chưa được trả lời

Chấm điểm của 1,00

Cờ câu hỏi

Lần lượt thêm các nút mang khóa khóa nguyên 53, 5, 66, 56, 97, 90, 41, 99, 74, 52, 22, 15 vào cây BST ban đầu rỗng. Mức (level) có số nút nhiều nhất là

- ☐ a. 2
- ☐ b. 3
- ☐ c. 4
- ☐ d. 1

Thời gian còn lại 0:55:09

Câu hỏi 4

Chưa được trả lời

Chấm điểm của 1,00

Cờ câu hỏi

Cho các cấu trúc dữ liệu của một danh sách liên kết đôi được khai báo trên ngôn ngữ C++ như sau:

```
class CNode {  
public:  
    int real, img;  
    Node* next;  
    Node* prev;  
};
```

Cho một danh sách liên kết đôi bao gồm các phần tử: (real = 15, img = -1), (real = 8, img = 2), (real = 5, img = -8), (real = 6, img = 3) và một con trỏ **t** đang giữ địa chỉ của phần tử (real = 8, img = 2).

Để truy cập được thành phần **img** của phần tử cuối cùng, ta sử dụng biểu thức

Lưu ý: không có khoảng trắng trong biểu thức và số lượng -> và . phải là ít nhất.

Thời gian còn lại 0:53:29

Câu hỏi 5

Chưa được trả lời

Chấm điểm của 1,00

Cờ câu hỏi

Phát biểu nào sau đây đúng khi nói về danh sách?

Chọn một hoặc nhiều hơn:

- ☐ a. Hàng đợi là một cấu trúc dữ liệu thể hiện của danh sách theo kiểu LIFO.
- ☐ b. Trong danh sách hạn chế, các thao tác thêm và xóa chỉ diễn ra tại một số vị trí nhất định của danh sách.
- ☐ c. Danh sách được chia thành hai loại: danh sách tổng quát và danh sách bị giới hạn
- ☐ d. Danh sách liên kết là một thể hiện của danh sách hạn chế.

Thời gian còn lại 0:52:26

Câu hỏi 6

Chưa được trả lời

Chấm điểm của 1,00

Cờ câu hỏi

Giả định rằng ta sử dụng Quick sort để sắp xếp một mảng 8 số nguyên, sau khi thực hiện partition lần đầu tiên, ta thu được mảng sau đây:

45 47 44 50 52 55 54 53

Pivot đã được chọn có giá trị là (nếu có nhiều đáp án, chỉ chọn viết 1 trong các đáp án)

Câu trả lời:

| I

Thời gian còn lại 0:51:22

Câu hỏi 7

Chưa được trả lời

Chấm điểm của 1,00

Cờ câu hỏi

Một cây nhị phân có:

- Hậu thứ tự là 23, 29, 33, 31, 25, 16.
- Trung thứ tự là 23, 16, 29, 25, 33, 31.

Tổng các nút trên mức (level) 0 và 1 của cây là

Câu trả lời:

| I

Câu hỏi 8

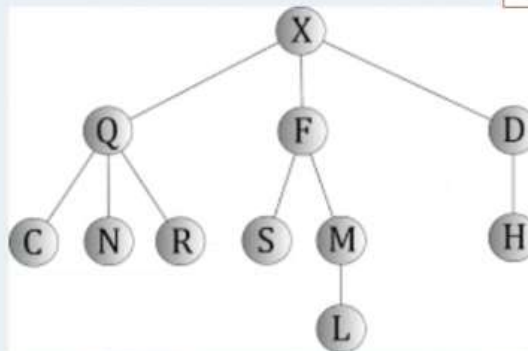
Chưa được trả lời

Chấm điểm của 1,00

Cờ câu hỏi

Cho cây như hình vẽ:

Thời gian còn lại 0:51:15



1. Bậc (degree) của nút F là , trong đó bậc ngoại (outdegree) là

2. Mức (level) 2 bao gồm các nút (viết mỗi nút cách nhau bởi một dấu phẩy, không có khoảng trắng trong đáp án).

3. Chiều cao của cây là .

Tất cả các đáp án đều không có khoảng trắng.

Thời gian còn lại 0:50:26

Câu hỏi 9

Chưa được trả lời

Chấm điểm của 1,00

🚩 Cờ câu hỏi

Khi sử dụng một mảng để hiện thực queue, độ phức tạp về thời gian khi thêm một phần tử vào queue:

Chọn một:

- ☐ a. $O(1)$
- ☐ b. $O(N \log N)$
- ☐ c. $O(N)$
- ☐ d. $O(N^2)$

Thời gian còn lại 0:49:46

Câu hỏi 10

Chưa được trả lời

Chấm điểm của 1,00

🚩 Cờ câu hỏi

```
int a = 0, i = 1;
while (i < N) {
    a += i;
    i *= 3;
}
```

Độ phức tạp của đoạn code trên theo N được biểu diễn theo Big-O là:

Chọn một:

- ☐ a. $O(N^2 \log N)$
- ☐ b. $O(\log N)$
- ☐ c. $O(N)$
- ☐ d. $O(\sqrt{N})$

Thời gian còn lại 0:48:30

Câu hỏi 11

Chưa được trả lời

Chấm điểm của 1,00

Cờ câu hỏi

Hãy điền vào các chỗ trống dưới đây để hoàn thành việc khai báo các cấu trúc dữ liệu phù hợp để lưu trữ một danh sách liên kết đơn trong đó mỗi phần tử lưu trữ thông tin của một số phức (bao gồm có hai thành phần: *real* - phần thực, *img* - phần ảo):

```
class Complex { public: double real, img; };
```

```
class Node {
```

```
public:
```

```
Complex value;
```

```
| next;
```

```
};
```

```
class CList {
```

```
| head;
```

```
};
```

Lưu ý: không tồn tại khoảng trắng trong các đáp án trên.

Thời gian còn lại 0:48:07

Câu hỏi 12

Chưa được trả lời

Chấm điểm của 1,00

Cờ câu hỏi

Cho một hàm đệ quy được hiện thực trong ngôn ngữ C/C++ như sau:

```
int rec(int n) {  
    if (n == 0) return 2;  
    if (n == 1) return 3;  
    return 3 * rec(n - 1) + 4 * rec(n - 2);  
}
```

Lời gọi `rec(6)` trả về giá trị:

Câu trả lời:

Thời gian còn lại 0:47:29

Câu hỏi 13

Chưa được trả lời

Chấm điểm của 1,00

Cờ câu hỏi

Giá trị của biểu thức toán học được viết dưới dạng tiền tố $* + 2\ 8 - 8\ 5$ là

Câu trả lời:

Thời gian còn lại 0:47:21

Câu hỏi 14

Chưa được trả lời

Chấm điểm của 1,00

Cờ câu hỏi

Trong một trạng thái của quá trình sắp xếp bằng giải thuật Merge sort, mảng số nguyên được chia thành hai phần đã sắp xếp để tiến hành merge như sau: [37, 38, 39 | 31, 32, 33].

Chi tiết của quá trình **merge** được thể hiện ở mã giả dưới đây:

```
// copy to aux[]
for (int k = lo; k <= hi; k++) {
    aux[k] = a[k];
}

// merge back to a[]
int i = lo, j = mid+1;
for (int k = lo; k <= hi; k++) {
    if (i > mid) a[k] = aux[j++];
    else if (j > hi) a[k] = aux[i++];
    else if (less(aux[j], aux[i])) a[k] = aux[j++];
    else a[k] = aux[i++];
}
```

Số phép so sánh trên các phần tử của mảng phải thực hiện để merge hai phần này là

Câu trả lời:

1248?cmid=8801&page=13#

Thời gian còn lại 0:46:44

Câu hỏi 15

Chưa được trả lời

Chấm điểm của 1,00

Cờ câu hỏi

```
int a = 0, i = N;
while (i > 0) {
    a += i;
    i /= 2;
}
```

Độ phức tạp của đoạn code trên theo N được biểu diễn theo Big-O là:

Chọn một:

- ☐ a. $O(N)$
- ☐ b. $O(\sqrt{N})$
- ☐ c. $O(N^2 \log N)$
- ☐ d. $O(\log N)$

Thời gian còn lại 0:44:45

Câu hỏi 16

Chưa được trả lời

Chấm điểm của 1,00

Cờ câu hỏi

Đâu là lợi thế của Selection Sort so với các kỹ thuật sắp xếp khác?

Chọn một hoặc nhiều hơn:

- ☐ a. Không đòi hỏi thêm không gian vùng nhớ
- ☐ b. Nó làm nhanh hơn các kỹ thuật sắp xếp khác
- ☐ c. Nó có thể mở rộng với tập dữ liệu lớn
- ☐ d. Nó làm việc tốt trên dữ liệu đã được sắp xếp

Câu hỏi 17

Chưa được trả lời

Chấm điểm của 1,00

🚩 Cờ câu hỏi

Thời gian còn lại 0:44:27

Cho mảng các số nguyên [24, 22, 26, 19, 21].

Chỉ phí cần phải trả để số **19** đứng đúng vị trí của nó trong mảng đã được sắp xếp bằng **giải thuật Bubble Sort**. Biết rằng, một lần trao đổi (swap) giữa hai phần tử tốn chỉ phí là 11 đồng.

Chi tiết giải thuật Bubble Sort đã sử dụng được cho như mã giả dưới đây:

```

1 curr = 0
2 flag = False
3 while curr < count AND flag = False do
4     step = count - 1
5     flag = True
6     while step > curr do
7         if data[step] < data[step - 1] then
8             flag = False
9             swap(data[step], data[step - 1])
10        end
11        step = step - 1
12    end
13    curr = curr + 1
14 end
    
```

Câu trả lời:

Câu hỏi 18

Chưa được trả lời

Chấm điểm của 1,00

🚩 Cờ câu hỏi

Thời gian còn lại 0:42:52

Cho mảng các số nguyên: 4 3 2 10 12 1 5 6.

Sử dụng **Insertion Sort** để thực hiện sắp xếp mảng trên với **thứ tự tăng dần**.

Đâu có thể là một trạng thái của mảng có khi vừa chèn một phần tử nào đó vào đúng vị trí của nó?

Chi tiết giải thuật Insertion Sort đã sử dụng được thể hiện ở phần mã giả dưới đây:

```

1 if count > 1 then
2     curr = 1
3     while curr < count do
4         tmp = data[curr]
5         step = curr - 1
6         while step ≥ 0 AND tmp < data[step] do
7             data[step + 1] = data[step]
8             step = step - 1
9         end
10        data[step + 1] = tmp
11        curr = curr + 1
12    end
13 end
    
```

Chọn một:

- ☐ a. 4 2 3 10 12 1 5 6
- ☐ b. 1 3 4 10 12 2 5 6
- ☐ c. 2 3 4 6 12 1 5 10
- ☐ d. 2 3 4 10 12 1 5 6

Thời gian còn lại 0:41:03

Câu hỏi **19**

Chưa được trả lời

Chấm điểm của 1,00

🚩 Cờ câu hỏi

Cho một cây nhị phân được viết dưới dạng sử dụng cặp ngoặc (parenthesis listing) như sau:

15 (16 (9 (8))) (21 (22) (16))

Người ta sử dụng **hàng đợi** (queue) để **duyệt theo chiều rộng** (breadth-first search) cây nhị phân nói trên. Trong quá trình duyệt, **tại thời điểm hàng đợi có nhiều phần tử nhất thì tổng các giá trị của các phần tử có trong hàng đợi** là

Câu trả lời:

Câu hỏi **20**

Chưa được trả lời

Chấm điểm của 1,00

🚩 Cờ câu hỏi

Cho một hiện thực danh sách liên kết đôi như sau:

Thời gian còn lại 0:38:37

```
class Node
```

```
{
```

```
public:
```

```
    int value; Node *prev, *next;
```

```
    Node(int value = 0, Node *prev = NULL, Node *next = NULL) : value(value), prev(prev),  
    next(next) {}
```

```
};
```

```
class DLinkedList
```

```
{
```

```
private:
```

```
    Node *head, *tail; // to store the address of the first and last node
```

```
    int count; // to store the length of list
```

```
public:
```

```
    DLinkedList() : head(NULL), tail(NULL), count(0) {}
```

```
    int get(int index)
```

```
{
```

```
    if (index < 0 || index >= this->count)
```

```
        throw std::out_of_range("Index is out of range");
```

```
    Node* temp = NULL;
```

```
    if (index < this->count / 2) {
```

```

temp = this->head;
for (int i = 0; i < index; i++)
    temp = temp->next;
}
else {
    temp = this->tail;
    for (int i = ; i > ; i--)
        temp = ;
}
return temp->value;
}
};

```

Biết rằng, **get** là phương thức để lấy và trả về giá trị của phần tử tại vị trí **index** (bắt đầu từ 0).
 Hãy điền vào các ô trống để hoàn thành đoạn mã trên.

Thời gian còn lại 0:38:35

Câu hỏi **21**

Chưa được trả lời

Chấm điểm của 1,00

🚩 Cờ câu hỏi

Cho các khai báo và hiện thực của một danh sách liên kết đơn như sau:

```

class Node {
public:
    int value; Node* next;
    Node(int value = 0) : value(value) {};
};

class SLinkedList {
private:
    Node* head;
public:
    void prepend(int value) {
        if (  )
            head = new Node(value);
        else {
            Node* temp = new Node(value);
             = head;
            
        }
    }
};

```

Biết rằng, **prepend** là phương thức dùng để thêm một phần tử vào đầu danh sách.

```

void prepend(int value) {
    if ( ) {
        head->value == NULL
        head->next == NULL
        head == NULL
    }
    else {
        Node* temp = new Node(value);
        = head;
        temp->next
        temp->value
        temp
    }
}

```

Thời gian còn lại 0:38:10

Câu hỏi 22

Chưa được trả lời

Chấm điểm của 5,00

Cờ câu hỏi

Để lưu trữ các **điểm trong không gian 3 chiều** (bao gồm các tọa độ x, y, z), người ta dùng một danh sách liên kết đơn.

Sinh viên sử dụng ngôn ngữ lập trình C/C++ để:

- Khái báo các cấu trúc dữ liệu phù hợp để mô tả danh sách liên kết đơn các điểm trong không gian 3 chiều.
- Hiện thực hàm hoặc phương thức phù hợp với cách khai báo ở câu trên để in ra điểm đầu tiên có khoảng cách Euclidean đến gốc tọa độ $O(0, 0, 0)$ là lớn nhất.

Sinh viên chỉ được phép sử dụng các kiểu dữ liệu cơ bản (primitive types) của C/C++ để hiện thực các yêu cầu nêu trên.