TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH

Đề thi số: 1

I. TRẮC NGHIỆM (3 điểm)

(D) Singly-linked lists hoặc Doubly-linked lists đều

MSSV:

phù hợp

Họ tên:

Môn thi: Cấu trúc dữ liệu và giải thuật

viện, các hàm cơ bản và khai báo khác (nếu cần thiết) được xem như đầy đủ

 \square Không được phép dùng tài liệu

Нọ	và	tê	n:	 	 	 	 	 	 		 			
MS	SSV	⁷ :		 	 	 	 	 	 		 		 	

Trang 1/6 — Mã đề 1

Thời gian làm bài: 120 phút

 \boxtimes Được phép dùng tài liệu giấy

Đề thi gồm 6 trang.

KIỂM TRA CUỐI KỲ (25/07/2020)

Chú ý: Sinh viên làm trực tiếp trên đề thi; mỗi câu hỏi trắc nghiệm chỉ chọn một đáp án đúng nhất. Các khai báo về thư

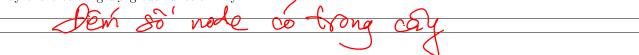
Câu 1. Cho array A chứa n số nguyên có giá trị khác nhau từng đôi một. Giả sử muốn tìm giá trị nhỏ nhất trong A, trong trường hợp xấu nhất, giải pháp nào sau đây hiệu quả nhất: (A) Sort A tăng dần, min là phần tử đầu tiên (B) Lặp qua cả dãy tìm phần tử min (C) Tạo 1 BST từ A, tìm phần tử min	Câu 7. Thuật ngữ nào dùng để diễn tả giải thuật O(n): (A) Constant (B) Linear (C) Logarithmic (D) Cubic Câu 8. Trong trường hợp tốt nhất, giải thuật sort nào hiệu quả nhất: (B) Quick sort
D Tạo 1 MinHeap từ A, min được chứa trong root	© Merge sort D Selection sort
Câu 2. Chọn phát biểu sai (A) Linear Searching có độ phức tạp O(n) (B) Binary Searching trên Ordered Array có độ phức tạp O(log n) (C) Searching trên cây AVL có độ phức tạp O(ln n)	Câu 9. Số node tối thiểu của 1 full binary tree có depth = 3 (cây chỉ có root xem như có depth = 0) là: 7 B 3 C 5 D 15 E 8
Searching trên Heap có độ phức tạp O(log n)	Câu 10. Con trỏ (pointers gồm front, rear) nào sẽ bị thay đổi khi insert 1 phần tử vào 1 queue đang có 1 phần
Câu 3. Giả sử cần chuyển biểu thức trung tố sang hậu tố, cấu trúc dữ liệu nào phù hợp nhất được sử dụng để giải quyết bài toán này: (A) Queue (B) List (C) Tree (Stack)	tử: (A) front (C) cả 2 (D) không có con trỏ nào thay đổi (Câu 11. 1 cây nhị phân khi duyệt NLR cho kết quả 14 2 1
Câu 4. Trong trường hợp tốt nhất, giải thuật sort nào kém hiệu quả nhất: (A) Insertion sort (B) Quick sort (C) Bubble sort (D) Selection sort	3 11 10 7 30 40, duyệt LRN cho kết quả 1 3 2 7 10 40 30 11 14, duyệt LNR sẽ có kết quả là: (A) 1 2 3 7 10 11 14 30 40 (B) 1 2 3 14 7 10 11 40 30 (C) 1 2 3 7 10 14 30 11 40 (D) 14 1 2 3 10 11 7 40 30
Câu 5. Xét đoạn code sau, độ phức tạp Big-O: int test = 0, i, j; for (i = 0; i < n; i++): for (j = 0; j < i; j++): test = test + i * j; A O(n) O(n ²)	Câu 12. 1 hash table có length 10 sử dụng hash function h(k) = k mod 10, và linear probling. Sau khi chèn 6 giá trị vào 1 empty hash table, thì table chứa giá trị như sau: [_,_, 42, 23, 34, 52, 46, 33, _,_ (index của table được đánh từ 0). Các key value được chèn theo thứ tự trước sau như thế nào? (A) 46, 42, 34, 52, 23, 33 (B) 34, 42, 23, 52, 33, 46
\bigcirc O(log n) \bigcirc D O(1)	46, 34, 42, 23, 52, 33 D 42, 46, 33, 23, 34, 52 Câu 13. Xét giải thuật dùng stack để xác định sự cân bằng
Câu 6. Dạng nào của List phù hợp để trả lời câu hỏi "Phần tử thứ i của list chứa gì?": Lists hiện thực bằng 1 array B Singly-linked lists	của 1 chuỗi các dấu đóng, mở ngoặc. Số phầi tử trong stack tại một thời điểm (AT ANY ONI TIME) lớn nhất là bao nhiêu khi phân tích chuỗ sau: (()(())(()))
(C) Doubly-linked lists (danh sách liên kết 2 chiều)	(A) 5 hoặc nhiều hơn (B) 1

II. ĐIỀN KẾT QUẢ (4 điểm)

Câu 14. Cho function sau:

```
int mystery_2(Node *tree = this.root):
    if(tree == nullptr)
        return 0;
    else if (tree.getLeft() == nullptr and tree.getRight() == nullptr):
        return 1;
    return 1 + mystery_2(tree.getLeft()) + mystery_2(tree.getRight());
```

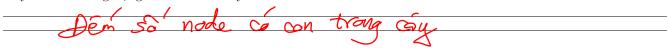
Hãy cho biết công dụng của function này:



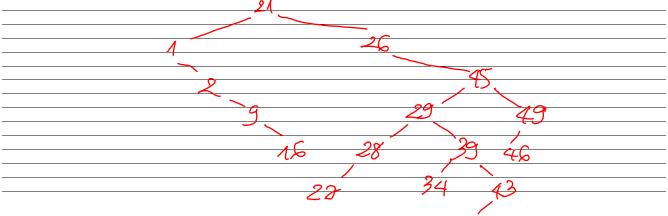
Câu 15. Cho function sau:

```
int mystery_1(Node *tree = this.root):
    if(tree == nullptr):
        return 0;
    elif (tree.getLeft() == nullptr and tree.getRight() == nullptr):
        return 0;
    return 1 + mystery 1(tree.getLeft()) + mystery 1(tree.getRight());
```

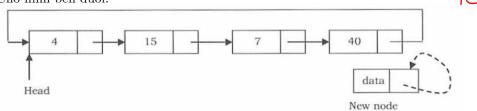
Hãy cho biết công dụng của function này:



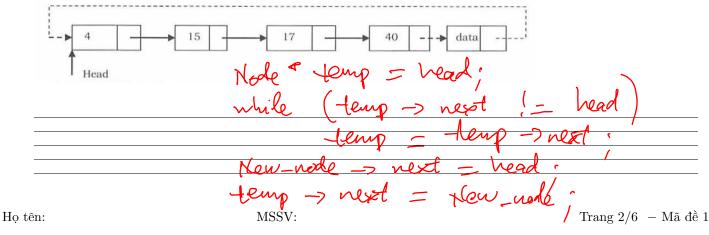
Câu 16. Cho array A = [21, 1, 26, 45, 29, 28, 2, 9, 16, 49, 39, 27, 43, 34, 46, 40], lần lượt insert các phần tử của A theo thứ tự index từ <math>0->15 vào 1 empty BST, hãy vẽ BST cuối cùng được tạo ra.

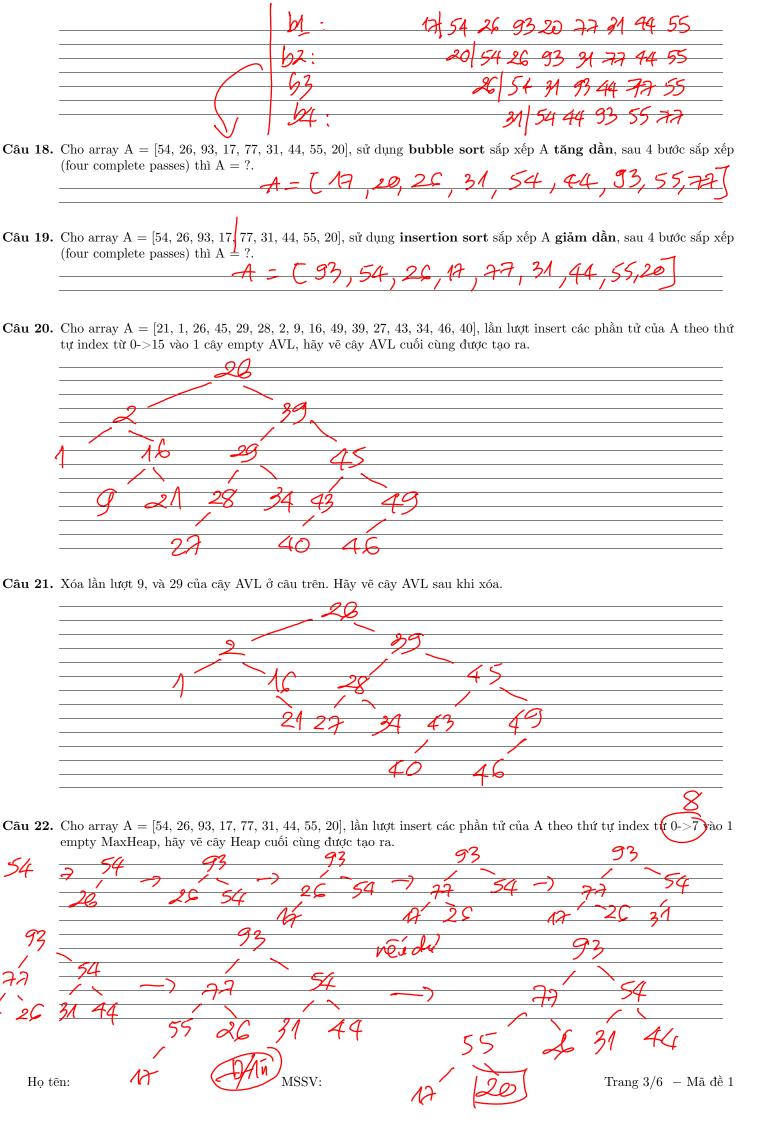


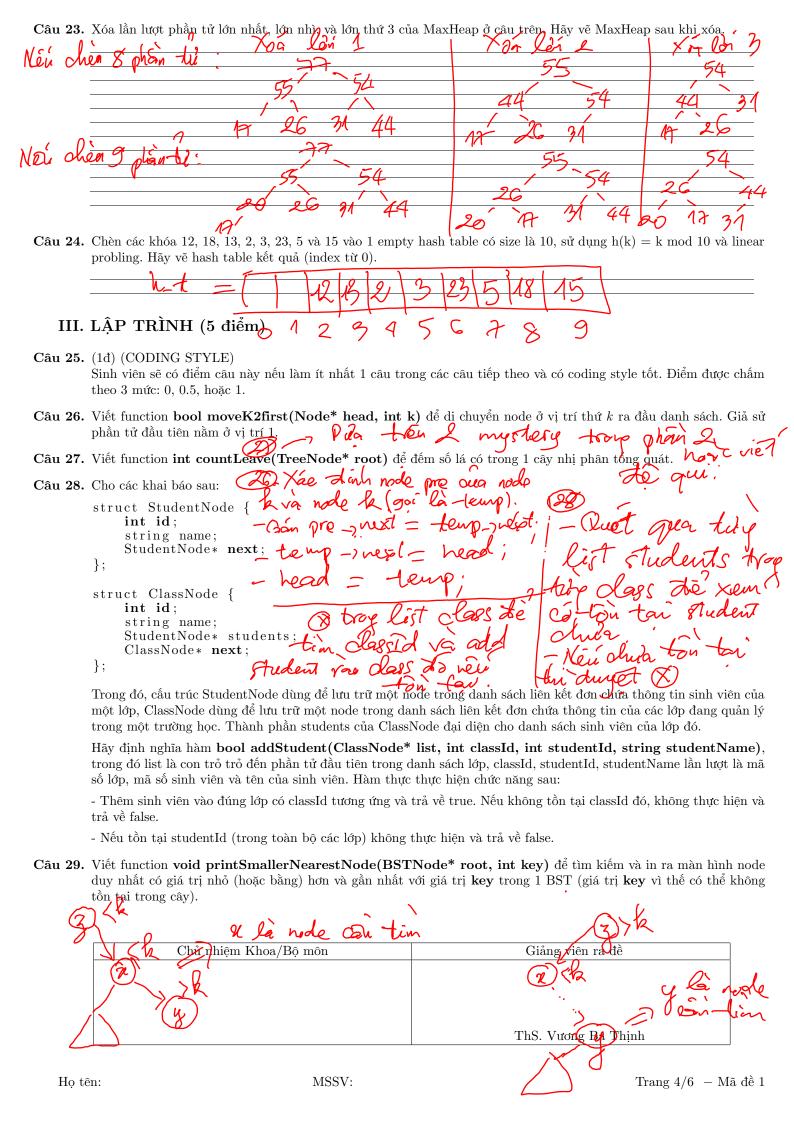
Câu 17. Cho hình bên dưới:



Hãy viết các câu lệnh cần thiết để có thể chèn phần tử New node vào vị trí như hình sau, giả sử đã có 2 con trỏ Head và New node như hình:







 \square Không được phép dùng tài liệu

Đề thi số: 1

Họ tên:

Môn thi: Cấu trúc dữ liệu và giải thuật

KIỂM TRA CUỐI KỲ (25/07/2020)

Thời gian làm bài: 120 phút

⊠ Được phép dùng tài liệu giấy

I. TRẮC NGHIỆM (3 điểm)

Câu 1. Cho array A chứa n số nguyên có giá trị khác nhau từng đôi một. Giả sử muốn tìm giá trị nhỏ nhất	Câu 7. Thuật ngữ nào dùng để diễn tả giải thuật O(n):
trong A, trong trường hợp xấu nhất, giải pháp nào	A Constant B Linear
sau đây hiệu quả nhất:	C Logarithmic D Cubic
A Sort A tăng dần, min là phần tử đầu tiên B Lặp qua cả dãy tìm phần tử min	Câu 8. Trong trường hợp tốt nhất, giải thuật sort nào hiệ quả nhất:
(C) Tạo 1 BST từ A, tìm phần tử min	A Insertion sort (B) Quick sort
(D) Tao 1 MinHeap từ A, min được chứa trong root	(C) Merge sort (D) Selection sort
Câu 2. Chọn phát biểu sai	Câu 9. Số node tối thiểu của 1 full binary tree có dept = 3 (cây chỉ có root xem như có depth = 0) là:
(A) Linear Searching có độ phức tạp O(n)	
(B) Binary Searching trên Ordered Array có độ	(B) 3 (C) 5 (D) 15
phức tạp $O(\log n)$	(C) 5 (E) 8
(C) Searching trên cây AVL có độ phức tạp O(ln	
n) Searching trên Heap có độ phức tạp $O(\log n)$	Câu 10. Con trỏ (pointers gồm front, rear) nào sẽ bị thạ đổi khi insert 1 phần tử vào 1 queue đang có 1 phầ tử:
Câu 3. Giả sử cần chuyển biểu thức trung tố sang hậu tố, cấu trúc dữ liệu nào phù hợp nhất được sử dụng để giải quyết bài toán này:	(A) front (B) rear (C) cå 2
	D không có con trỏ nào thay đổi
(A) Queue (B) List (C) Tree (D) Stack	Câu 11. 1 cây nhị phân khi duyệt NLR cho kết quả 14 2
D Stack	3 11 10 7 30 40 , duyệt LRN cho kết quả 1 3 2
Câu 4. Trong trường hợp tốt nhất, giải thuật sort nào kém	10 40 30 11 14 , duyệt LNR sẽ có kết quả là:
hiệu quả nhất:	(A) 1 2 3 7 10 11 14 30 40
(A) Insertion sort (B) Quick sort	B 1 2 3 14 7 10 11 40 30
© Bubble sort © Selection sort	C 1 2 3 7 10 14 30 11 40
	(D) 14 1 2 3 10 11 7 40 30
Câu 5. Xét đoạn code sau, độ phức tạp Big-O:	Câu 12. 1 hash table có length 10 sử dụng hash function
$\begin{array}{ll} \mathbf{int} & \mathbf{test} = 0, \ \mathbf{i}, \ \mathbf{j}; \\ \mathbf{for} & (\mathbf{i} = 0), \ \mathbf{i} < n, \ \mathbf{i} + 1, \end{array}$	h(k) = k mod 10, và linear probling. Sau khi chè 6 giá trị vào 1 empty hash table, thì table chứa gi
for $(i = 0; i < n; i++):$ for $(j = 0; j < i; j++):$	trị như sau: [_, _, 42, 23, 34, 52, 46, 33, _, _
test = test + i * j;	(index của table được đánh từ 0). Các key value
	được chèn theo thứ tự trước sau như thế nào?
(A) O(n)	(A) 46, 42, 34, 52, 23, 33 (B) 34, 42, 23, 52, 33, 46
\bigcirc O(log n) \bigcirc D O(1)	A 46, 42, 34, 52, 23, 33 B 34, 42, 23, 52, 33, 46 C 46, 34, 42, 23, 52, 33 D 42, 46, 33, 23, 34, 52
Câu 6. Dạng nào của List phù hợp để trả lời câu hỏi "Phần	Câu 13. Xét giải thuật dùng stack để xác định sự cân bằn
tử thứ i của list chứa gì?":	của 1 chuỗi các dấu đóng, mở ngoặc. Số phầ
A Lists hiện thực bằng 1 array	tử trong stack tại một thời điểm (AT ANY ONI TIME) lớn nhất là bao nhiêu khi phân tích chuẩ
	sau: $(()(())(()))$
B Singly-linked lists C Doubly-linked lists (danh sách liên kết 2 chiều)	A 5 hoặc nhiều hơn (B) 1
D Singly-linked lists hoặc Doubly-linked lists đều	(C) 2 (D) 3
phù hợp	(E) 4

Trang 1/6 — Mã đề 1

MSSV:

II. ĐIỀN KẾT QUẢ (4 điểm)

Câu 14. Cho function sau:

```
int mystery_2(Node *tree = this.root):
    if(tree == nullptr)
        return 0;
    else if (tree.getLeft() == nullptr and tree.getRight() == nullptr):
        return 1;
    return 1 + mystery_2(tree.getLeft()) + mystery_2(tree.getRight());
```

Hãy cho biết công dụng của function này:

Câu 15. Cho function sau:

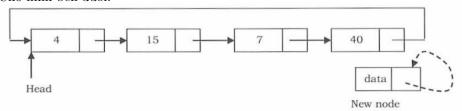
```
int mystery_1(Node *tree = this.root):
    if(tree == nullptr):
        return 0;
    elif (tree.getLeft() == nullptr and tree.getRight() == nullptr):
        return 0;
    return 1 + mystery_1(tree.getLeft()) + mystery_1(tree.getRight());
```

Hãy cho biết công dụng của function này:

Câu 16. Cho array A = [21, 1, 26, 45, 29, 28, 2, 9, 16, 49, 39, 27, 43, 34, 46, 40], lần lượt insert các phần tử của A theo thứ tự index từ <math>0->15 vào 1 empty BST, hãy vẽ BST cuối cùng được tạo ra.



Câu 17. Cho hình bên dưới:



Hãy viết các câu lệnh cần thiết để có thể chèn phần tử New node vào vị trí như hình sau, giả sử đã có 2 con trỏ Head và New node như hình:

Head	15	17	4 0 - → data	a	

Câu 18.	Cho array $A = [54, 26, 93, 17, 77, 31, 44, 55, 20]$, sử dụng bubble sort sắp xếp A tăng dần , sau 4 bước sắp xếp (four complete passes) thì $A = ?$.
Câu 19.	Cho array $A = [54, 26, 93, 17, 77, 31, 44, 55, 20]$, sử dụng insertion sort sắp xếp A giảm dần , sau 4 bước sắp xếp (four complete passes) thì $A = ?$.
Câu 20.	Cho array $A = [21, 1, 26, 45, 29, 28, 2, 9, 16, 49, 39, 27, 43, 34, 46, 40]$, lần lượt insert các phần tử của A theo thứ tự index từ $0->15$ vào 1 cây empty AVL , hãy vẽ cây AVL cuối cùng được tạo ra.
Câu 21.	Xóa lần lượt 9, và 29 của cây AVL ở câu trên. Hãy vẽ cây AVL sau khi xóa.
Câu 22.	Cho array $A = [54, 26, 93, 17, 77, 31, 44, 55, 20]$, lần lượt insert các phần tử của A theo thứ tự index từ $0->7$ vào 1 empty MaxHeap, hãy vẽ cây Heap cuối cùng được tạo ra.

 ${f Câu}$ 23. Xóa lần lượt phần tử lớn nhất, lớn nhì, và lớn thứ 3 của MaxHeap ở câu trên. Hãy vẽ MaxHeap sau khi xóa.

-	
· ·	
ii -	

Câu 24. Chèn các khóa 12, 18, 13, 2, 3, 23, 5 và 15 vào 1 empty hash table có size là 10, sử dụng $h(k) = k \mod 10$ và linear probling. Hãy vẽ hash table kết quả (index từ 0).

III. LẬP TRÌNH (5 điểm)

Câu 25. (1d) (CODING STYLE)

Sinh viên sẽ có điểm câu này nếu làm ít nhất 1 câu trong các câu tiếp theo và có coding style tốt. Điểm được chấm theo 3 mức: 0, 0.5, hoặc 1.

- Câu 26. Viết function bool moveK2first(Node* head, int k) để di chuyển node ở vị trí thứ k ra đầu danh sách. Giả sử phần tử đầu tiên nằm ở vị trí 1.
- Câu 27. Viết function int countLeave(TreeNode* root) để đếm số lá có trong 1 cây nhị phân tổng quát.
- Câu 28. Cho các khai báo sau:

```
struct StudentNode {
    int id;
    string name;
    StudentNode* next;
};

struct ClassNode {
    int id;
    string name;
    StudentNode* students;
    ClassNode* next;
};
```

Trong đó, cấu trúc StudentNode dùng để lưu trữ một node trong danh sách liên kết đơn chứa thông tin sinh viên của một lớp, ClassNode dùng để lưu trữ một node trong danh sách liên kết đơn chứa thông tin của các lớp đang quản lý trong một trường học. Thành phần students của ClassNode đại diện cho danh sách sinh viên của lớp đó.

Hãy định nghĩa hàm bool addStudent(ClassNode* list, int classId, int studentId, string studentName), trong đó list là con trỏ trỏ đến phần tử đầu tiên trong danh sách lớp, classId, studentId, studentName lần lượt là mã số lớp, mã số sinh viên và tên của sinh viên. Hàm thực thực hiện chức năng sau:

- Thêm sinh viên vào đúng lớp có class Id tương ứng và trả về true. Nếu không tồn tại class Id đó, không thực hiện và trả về false.
- Nếu tồn tại studentId (trong toàn bộ các lớp) không thực hiện và trả về false.
- Câu 29. Viết function void printSmallerNearestNode(BSTNode* root, int key) để tìm kiếm và in ra màn hình node duy nhất có giá trị nhỏ (hoặc bằng) hơn và gần nhất với giá trị key trong 1 BST (giá trị key vì thế có thể không tồn tại trong cây).

Họ tên: Trang 4/6 – Mã đề 1

TRƯỜNG ĐẠI HỌC BÁCH KHOA	
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍN	Η

Họ và tên:	
MSSV:	

Trang 1/6 — Mã đề 3

KIỂM TRA CUỐI KỲ (25/07/2020)

won tini. Cau truc du neu va giai tinua	1 noi gian iam bai. 120 piiut
	⊠ Được phép dùng tài liệu giấy ề thi gồm 6 trang.
- · · · · · · · · · · · · · · · · · · ·	nghiệm chỉ chọn một đáp án đúng nhất. Các khai báo về thư c (nếu cần thiết) được xem như đầy đủ
I. TRẮC NGI	HIỆM (3 điểm)
 Câu 1. 1 hash table có length 10 sử dụng hash function h(k) = k mod 10, và linear probling. Sau khi chèn 6 giá trị vào 1 empty hash table, thì table chứa giá trị như sau: [_,_, 42, 23, 34, 52, 46, 33,,_] (index của table được đánh từ 0). Các key values được chèn theo thứ tự trước sau như thế nào? (A) 42, 46, 33, 23, 34, 52 (B) 46, 42, 34, 52, 23, 33 (C) 34, 42, 23, 52, 33, 46 (D) 46, 34, 42, 23, 52, 33 	Câu 8. Xét giải thuật dùng stack để xác định sự cân bằn của 1 chuỗi các dấu đóng, mở ngoặc. Số phầi tử trong stack tại một thời điểm (AT ANY ONI TIME) lớn nhất là bao nhiêu khi phân tích chuỗ sau: (()(())(())) (A) 5 hoặc nhiều hơn (B) 4 (C) 1 (D) 2
Câu 2. Xét đoạn code sau, độ phức tạp Big-O:	
$\begin{array}{lll} \textbf{int} & \text{test} = 0 , \; i , \; j ; \\ \textbf{for} & (i = 0; \; i < n; \; i++): \\ & \textbf{for} & (j = 0; \; j < i; \; j++): \\ & \text{test} = \text{test} + i * j ; \end{array}$	Câu 9. Trong trường hợp tốt nhất, giải thuật sort nào hiệ quả nhất: (A) Selection sort (B) Insertion sort (C) Quick sort (D) Merge sort
	S Quick Bott
Câu 3. Con trỏ (pointers gồm front, rear) nào sẽ bị thay đổi khi insert 1 phần tử vào 1 queue đang có 1 phần tử: (A) không có con trỏ nào thay đổi (B) front (C) rear (D) cả 2	 Câu 10. Cho array A chứa n số nguyên có giá trị khác nha từng đôi một. Giả sử muốn tìm giá trị nhỏ nhất trong A, trong trường hợp xấu nhất, giải pháp nà sau đây hiệu quả nhất: (A) Tạo 1 MinHeap từ A, min được chứa trong root (B) Sort A tăng dần, min là phần tử đầu tiên (C) Lặp qua cả dãy tìm phần tử min
 Câu 4. Giả sử cần chuyển biểu thức trung tố sang hậu tố, cấu trúc dữ liệu nào phù hợp nhất được sử dụng để giải quyết bài toán này: (A) Stack (B) Queue (C) List (D) Tree 	 (D) Tạo 1 BST từ A, tìm phần tử min Câu 11. 1 cây nhị phân khi duyệt NLR cho kết quả 14 2 3 3 11 10 7 30 40, duyệt LRN cho kết quả 1 3 2 3
Câu 5. Thuật ngữ nào dùng để diễn tả giải thuật O(n): (A) Cubic (B) Constant (C) Linear (D) Logarithmic	10 40 30 11 14, duyệt LNR sẽ có kết quả là: (A) 14 1 2 3 10 11 7 40 30 (B) 1 2 3 7 10 11 14 30 40 (C) 1 2 3 14 7 10 11 40 30 (D) 1 2 3 7 10 14 30 11 40
Câu 6. Số node tối thiểu của 1 full binary tree có depth = 3 (cây chỉ có root xem như có depth = 0) là: (A) 7 (B) 8 (C) 3 (D) 5 (E) 15	Câu 12. Chọn phát biểu sai A Searching trên Heap có độ phức tạp O(log n) B Linear Searching có độ phức tạp O(n) C Binary Searching trên Ordered Array có độ
Câu 7. Dạng nào của List phù hợp để trả lời câu hỏi "Phần tử thứ i của list chứa gì?": (A) Singly-linked lists hoặc Doubly-linked lists đều	phức tạp $O(\log n)$ \bigcirc Searching trên cây AVL có độ phức tạp $O(\ln n)$
phù hợp B Lists hiện thực bằng 1 array C Singly-linked lists D Doubly-linked lists (danh sách liên kết 2 chiều)	Câu 13. Trong trường hợp tốt nhất, giải thuật sort nào kén hiệu quả nhất: (A) Selection sort (B) Insertion sort (C) Quick sort (D) Bubble sort

MSSV:

Họ tên:

II. ĐIỀN KẾT QUẢ (4 điểm)

	empty MaxHeap, hãy vẽ cây Heap cuối cùng được tạo ra.
Câu 15.	Xóa lần lượt phần tử lớn nhất, lớn nhì, và lớn thứ 3 của MaxHeap ở câu trên. Hãy vẽ MaxHeap sau khi xóa.
Câu 16.	Cho hình bên dưới:
	4 15 7 40
	data
	Head New node Hãy viết các câu lệnh cần thiết để có thể chèn phần tử New node vào vị trí như hình sau, giả sử đã có 2 con trỏ
	Head và New node như hình:
	4 15 17 40 - data
	Head
Câu 17.	Cho array $A = [54, 26, 93, 17, 77, 31, 44, 55, 20]$, sử dụng insertion sort sắp xếp A giảm dần , sau 4 bước sắp xếp (four complete passes) thì $A = ?$.
Câu 18.	Cho array $A = [21, 1, 26, 45, 29, 28, 2, 9, 16, 49, 39, 27, 43, 34, 46, 40]$, lần lượt insert các phần tử của A theo thứ

tự index từ 0->15 vào 1 cây empty AVL, hãy vẽ cây AVL cuối cùng được tạo ra.

Câu 19.	Xóa lần lượt 9, và 29 của cây AVL ở câu trên. Hãy vẽ cây AVL sau khi xóa.
Câu 20.	Chèn các khóa 12, 18, 13, 2, 3, 23, 5 và 15 vào 1 empty hash table có size là 10, sử dụng h(k) = k mod 10 và linear probling. Hãy vẽ hash table kết quả (index từ 0).
Câu 21.	Cho function sau: int mystery 1 (Node *tree = this.root):
	<pre>if (tree == nullptr): return 0;</pre>
	<pre>elif (tree.getLeft() == nullptr and tree.getRight() == nullptr): return 0;</pre>
	<pre>return 1 + mystery_1(tree.getLeft()) + mystery_1(tree.getRight());</pre>
	Hãy cho biết công dụng của function này:
Câu 22.	Cho array $A = [54, 26, 93, 17, 77, 31, 44, 55, 20]$, sử dụng bubble sort sắp xếp A tăng dần , sau 4 bước sắp xếp (four complete passes) thì $A = ?$.
Câu 23 .	Cho function sau:
	<pre>int mystery_2(Node *tree = this.root): if(tree == nullptr) return 0;</pre>
	<pre>else if (tree.getLeft() == nullptr and tree.getRight() == nullptr): return 1;</pre>
	return 1 + mystery_2(tree.getLeft()) + mystery_2(tree.getRight());
	Hãy cho biết công dụng của function này:

III. LAP TRINH (5 điêm)

Câu 25. (1d) (CODING STYLE)

Sinh viên sẽ có điểm câu này nếu làm ít nhất 1 câu trong các câu tiếp theo và có coding style tốt. Điểm được chấm theo 3 mức: 0, 0.5, hoặc 1.

- Câu 26. Viết function bool moveK2first(Node* head, int k) để di chuyển node ở vị trí thứ k ra đầu danh sách. Giả sử phần tử đầu tiên nằm ở vị trí 1.
- Câu 27. Viết function int countLeave(TreeNode* root) để đếm số lá có trong 1 cây nhị phân tổng quát.
- Câu 28. Cho các khai báo sau:

```
struct StudentNode {
    int id;
    string name;
    StudentNode* next;
};

struct ClassNode {
    int id;
    string name;
    StudentNode* students;
    ClassNode* next;
};
```

Trong đó, cấu trúc Student Node dùng để lưu trữ một node trong danh sách liên kết đơn chứa thông tin sinh viên của một lớp, Class Node dùng để lưu trữ một node trong danh sách liên kết đơn chứa thông tin của các lớp đang quản lý trong một trường học. Thành phần students của Class Node đại diện cho danh sách sinh viên của lớp đó.

Hãy định nghĩa hàm bool addStudent(ClassNode* list, int classId, int studentId, string studentName), trong đó list là con trỏ trỏ đến phần tử đầu tiên trong danh sách lớp, classId, studentId, studentName lần lượt là mã số lớp, mã số sinh viên và tên của sinh viên. Hàm thực thực hiện chức năng sau:

- Thêm sinh viên vào đúng lớp có classId tương ứng và trả về true. Nếu không tồn tại classId đó, không thực hiện và trả về false.
- Nếu tồn tại studentId (trong toàn bộ các lớp) không thực hiện và trả về false.
- Câu 29. Viết function void printSmallerNearestNode(BSTNode* root, int key) để tìm kiếm và in ra màn hình node duy nhất có giá trị nhỏ (hoặc bằng) hơn và gần nhất với giá trị key trong 1 BST (giá trị key vì thế có thể không tồn tại trong cây).

Chủ nhiệm Khoa/Bộ môn	Giảng viên ra đề
	ThS. Vương Bá Thịnh

Ho tên: MSSV: Trang 4/6 – Mã đề 3

KIỂM TRA CUỐI KỲ (25/07/2020)

	CA COOLIGI	(29/01/2020)
Môn thi: Cấu trúc dữ li	ệu và giải thuật	Thời gian làm bài: 120 phút
\Box Không được phép dùng t Đề thi số: 3	ài liệu ⊠ Đươ	ợc phép dùng tài liệu giấy
	I. TRẮC NGI	HIỆM (3 điểm)
Câu 1. 1 hash table có length 10 sử d h(k) = k mod 10, và linear prob 6 giá trị vào 1 empty hash table trị như sau: [,, 42, 23, 34, (index của table được đánh từ được chèn theo thứ tự trước sau (A) 42, 46, 33, 23, 34, 52 (B) 46, 4 (C) 34, 42, 23, 52, 33, 46 (D) 46, 3	oling. Sau khi chèn , thì table chứa giá , 52, 46, 33, _, _] 0). Các key values 1 như thế nào?	 Câu 7. Dạng nào của List phù hợp để trả lời câu hỏi "Phần tử thứ i của list chứa gì?": A Singly-linked lists hoặc Doubly-linked lists đều phù hợp B Lists hiện thực bằng 1 array C Singly-linked lists D Doubly-linked lists (danh sách liên kết 2 chiều) Câu 8. Xét giải thuật dùng stack để xác định sự cân bằng
Câu 2. Xét đoạn code sau, độ phức tạp int test = 0, i, j; for (i = 0; i < n; i++) for (j = 0; j < i; test = test + i):	của 1 chuỗi các dấu đóng, mở ngoặc. Số phần tử trong stack tại một thời điểm (AT ANY ONE TIME) lớn nhất là bao nhiêu khi phân tích chuỗi sau: (()(())(())) A 5 hoặc nhiều hơn B 4 C 1 D 2
		 E 3 Câu 9. Trong trường hợp tốt nhất, giải thuật sort nào hiệu quả nhất:
Câu 3. Con trỏ (pointers gồm front, re đổi khi insert 1 phần tử vào 1 qua tử: (A) không có con trỏ nào thay đổi (B) front (C) rear (D) cả 2		A Selection sort C Quick sort D Merge sort Câu 10. Cho array A chứa n số nguyên có giá trị khác nhau từng đôi một. Giả sử muốn tìm giá trị nhỏ nhất trong A, trong trường hợp xấu nhất, giải pháp nào sau đây hiệu quả nhất: A Tạo 1 MinHeap từ A, min được chứa trong root B Sort A tăng dần, min là phần tử đầu tiên
Câu 4. Giả sử cần chuyển biểu thức tru cấu trúc dữ liệu nào phù hợp r để giải quyết bài toán này:		C Lặp qua cả dãy tìm phần tử min D Tạo 1 BST từ A, tìm phần tử min
A Stack B Quet C List D Tree	ne	Câu 11. 1 cây nhị phân khi duyệt NLR cho kết quả 14 2 1 3 11 10 7 30 40, duyệt LRN cho kết quả 1 3 2 7 10 40 30 11 14, duyệt LNR sẽ có kết quả là:
Câu 5. Thuật ngữ nào dùng để diễn tả (A) Cubic (B) Cons (C) Linear (D) Loga		(A) 14 1 2 3 10 11 7 40 30 (B) 1 2 3 7 10 11 14 30 40 (C) 1 2 3 14 7 10 11 40 30 (D) 1 2 3 7 10 14 30 11 40 (Câu 12. Chọn phát biểu sai
Câu 6. Số node tối thiểu của 1 full bin = 3 (cây chỉ có root xem như có A 7 B 8 C 3 D 5	-	A Searching trên Heap có độ phức tạp O(log n) B Linear Searching có độ phức tạp O(n) C Binary Searching trên Ordered Array có độ phức tạp O(log n) D Searching trên cây AVL có độ phức tạp O(ln

Họ tên: MSSV: Trang 1/6 – Mã đề 3

			<u> </u>	
		II. ĐIEN K	${ m ET~QU\AA~(4~di m \acute{e}m)}$	
'âu 14.	Cho array A = [54, 26, 93, empty MaxHeap, hãy vẽ câ			A theo thứ tự index từ 0->7 vào
Câu 15.	Xóa lần lượt phần tử lớn n	hất, lớn nhì, và lớn thứ	3 của MaxHeap ở câu trên. Hã	y vẽ MaxHeap sau khi xóa.
Câu 16.	Cho hình bên dưới:			
	4 15	7 -	data data	
	Hãy viết các câu lệnh cần		New node ần tử New node vào vị trí như	ư hình sau, giả sử đã có 2 con tr
	Head và New node như hìn	n:	40 - data	
	Head			

 $\mathbf{Câu}$ 13. Trong trường hợp tốt nhất, giải thuật sort nào kém

(four complete passes) thì A = ?.

hiệu quả nhất:

Câu 18.	Cho array $A = [21, 1, 26, 45, 29, 28, 2, 9, 16, 49, 39, 27, 43, 34, 46, 40]$, lần lượt insert các phần tử của A theo thứ tự index từ $0->15$ vào 1 cây empty AVL , hãy vẽ cây AVL cuối cùng được tạo ra.				
Câu 19.	Xóa lần lượt 9, và 29 của cây AVL ở câu trên. Hãy vẽ cây AVL sau khi xóa.				
Câu 20.	Chèn các khóa 12, 18, 13, 2, 3, 23, 5 và 15 vào 1 empty hash table có size là 10, sử dụng $h(k) = k \mod 10$ và linear probling. Hãy vẽ hash table kết quả (index từ 0).				
Câu 21.	Cho function sau:				
	<pre>int mystery_1(Node *tree = this.root): if(tree == nullptr): return 0;</pre>				
	<pre>elif (tree.getLeft() == nullptr and tree.getRight() == nullptr): return 0;</pre>				
	return 1 + mystery_1(tree.getLeft()) + mystery_1(tree.getRight()); Hãy cho biết công dụng của function này:				
Câu 22 .	Cho array $A = [54, 26, 93, 17, 77, 31, 44, 55, 20]$, sử dụng bubble sort sắp xếp A tăng dần , sau 4 bước sắp xếp (four complete passes) thì $A = ?$.				
Câu 23.	Cho function sau:				
	<pre>int mystery_2(Node *tree = this.root): if(tree == nullptr)</pre>				
	<pre>return 0; else if (tree.getLeft() == nullptr and tree.getRight() == nullptr): return 1;</pre>				
	<pre>return 1 + mystery_2(tree.getLeft()) + mystery_2(tree.getRight());</pre>				

MSSV:

 ${\rm Ho}$ tên:

Trang 3/6~- Mã đề 3

Câu 24.	Cho array $A=[21,1,26,45,29,28,2,9,16,49,39,27,43,34,46,40]$, lần lượt insert các phần tử của A theo th tự index từ $0->15$ vào 1 empty BST, hãy vẽ BST cuối cùng được tạo ra.

III. LÂP TRÌNH (5 điểm)

Câu 25. (1d) (CODING STYLE)

Sinh viên sẽ có điểm câu này nếu làm ít nhất 1 câu trong các câu tiếp theo và có coding style tốt. Điểm được chấm theo 3 mức: 0, 0.5, hoặc 1.

- Câu 26. Viết function bool moveK2first(Node* head, int k) để di chuyển node ở vị trí thứ k ra đầu danh sách. Giả sử phần tử đầu tiên nằm ở vi trí 1.
- Câu 27. Viết function int countLeave(TreeNode* root) để đếm số lá có trong 1 cây nhi phân tổng quát.
- Câu 28. Cho các khai báo sau:

```
struct StudentNode {
    int id;
    string name;
    StudentNode* next;
};

struct ClassNode {
    int id;
    string name;
    StudentNode* students;
    ClassNode* next;
};
```

Hãy cho biết công dụng của function này:

Trong đó, cấu trúc Student Node dùng để lưu trữ một node trong danh sách liên kết đơn chứa thông tin sinh viên của một lớp, Class Node dùng để lưu trữ một node trong danh sách liên kết đơn chứa thông tin của các lớp đang quản lý trong một trường học. Thành phần students của Class Node đại diện cho danh sách sinh viên của lớp đó.

Hãy định nghĩa hàm bool addStudent(ClassNode* list, int classId, int studentId, string studentName), trong đó list là con trỏ trỏ đến phần tử đầu tiên trong danh sách lớp, classId, studentId, studentName lần lượt là mã số lớp, mã số sinh viên và tên của sinh viên. Hàm thực thực hiện chức năng sau:

- Thêm sinh viên vào đúng lớp có class Id tương ứng và trả về true. Nếu không tồn tại class Id đó, không thực hiện và trả về false.
- Nếu tồn tại studentId (trong toàn bộ các lớp) không thực hiện và trả về false.
- Câu 29. Viết function void printSmallerNearestNode(BSTNode* root, int key) để tìm kiếm và in ra màn hình node duy nhất có giá trị nhỏ (hoặc bằng) hơn và gần nhất với giá trị key trong 1 BST (giá trị key vì thế có thể không tồn tại trong cây).

Họ tên: Trang 4/6 – Mã đề 3

TRƯỜNG ĐẠI HỌC BÁCH KHOA	
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH	ſ

Họ và tên:	
MSSV:	

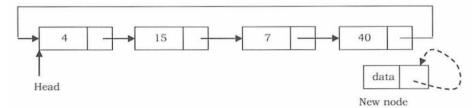
KIỂM TRA CUỐI KỲ (25/07/2020)

KIEM TRA CUUI	KY (25/07/2020)
Môn thi: Cấu trúc dữ liệu và giải thu	ật Thời gian làm bài: 120 phút
	☑ Được phép dùng tài liệu giấy Dề thi gồm 6 trang. c nghiệm chỉ chọn một đáp án đúng nhất. Các khai báo về thư
	ác (nếu cần thiết) được xem như đầy đủ
I. TRẮC NG	HIỆM (3 điểm)
 Câu 1. Giả sử cần chuyển biểu thức trung tố sang hậu tố, cấu trúc dữ liệu nào phù hợp nhất được sử dụng để giải quyết bài toán này: (A) Queue (B) Stack (C) List (D) Tree Câu 2. Xét đoạn code sau, độ phức tạp Big-O: 	Câu 7. 1 hash table có length 10 sử dụng hash function h(k) = k mod 10, và linear probling. Sau khi chèn 6 giá trị vào 1 empty hash table, thì table chứa giá trị như sau: [_,_, 42, 23, 34, 52, 46, 33, _,_] (index của table được đánh từ 0). Các key values được chèn theo thứ tự trước sau như thế nào? (A) 46, 42, 34, 52, 23, 33 (B) 42, 46, 33, 23, 34, 52 (C) 34, 42, 23, 52, 33, 46 (D) 46, 34, 42, 23, 52, 33
int test = 0, i, j; for (i = 0; i < n; i++): for (j = 0; j < i; j++): test = test + i * j; (A) O(n) (B) O(log n)	 C 34, 42, 23, 52, 33, 46 D 46, 34, 42, 23, 52, 33 Câu 8. Xét giải thuật dùng stack để xác định sự cân bằng của 1 chuỗi các dấu đóng, mở ngoặc. Số phần tử trong stack tại một thời điểm (AT ANY ONE TIME) lớn nhất là bao nhiêu khi phân tích chuỗi sau: (()(())(())) A 5 hoặc nhiều hơn B 1 C 4
Câu 3. Trong trường hợp tốt nhất, giải thuật sort nào hiệu quả nhất: (A) Insertion sort (B) Selection sort (C) Quick sort (D) Merge sort (Câu 4. Dạng nào của List phù hợp để trả lời câu hỏi "Phần tử thứ i của list chứa gì?":	E 3 Câu 9. 1 cây nhị phân khi duyệt NLR cho kết quả 14 2 1 3 11 10 7 30 40, duyệt LRN cho kết quả 1 3 2 7 10 40 30 11 14, duyệt LNR sẽ có kết quả là: (A) 1 2 3 7 10 11 14 30 40 (B) 14 1 2 3 10 11 7 40 30 (C) 1 2 3 14 7 10 11 40 30 (D) 1 2 3 7 10 14 30 11 40
 A Lists hiện thực bằng 1 array B Singly-linked lists hoặc Doubly-linked lists đều phù hợp C Singly-linked lists D Doubly-linked lists (danh sách liên kết 2 chiều) 	Câu 10. Số node tối thiểu của 1 full binary tree có depth = 3 (cây chỉ có root xem như có depth = 0) là: (A) 7 (B) 3 (C) 8 (D) 5 (E) 15
 Câu 5. Con trỏ (pointers gồm front, rear) nào sẽ bị thay đổi khi insert 1 phần tử vào 1 queue đang có 1 phần tử: (A) front (B) không có con trỏ nào thay đổi (C) rear (D) cả 2 	Câu 11. Chọn phát biểu sai A Linear Searching có độ phức tạp O(n) B Searching trên Heap có độ phức tạp O(log n) C Binary Searching trên Ordered Array có độ phức tạp O(log n) D Searching trên cây AVL có độ phức tạp O(ln n)
 Câu 6. Cho array A chứa n số nguyên có giá trị khác nhau từng đôi một. Giả sử muốn tìm giá trị nhỏ nhất trong A, trong trường hợp xấu nhất, giải pháp nào sau đây hiệu quả nhất: A Sort A tăng dần, min là phần tử đầu tiên 	Câu 12. Trong trường hợp tốt nhất, giải thuật sort nào kém hiệu quả nhất: (A) Insertion sort (B) Selection sort (C) Quick sort (D) Bubble sort
B Tạo 1 MinHeap từ A, min được chứa trong root C Lặp qua cả dãy tìm phần tử min D Tạo 1 BST từ A, tìm phần tử min	Câu 13. Thuật ngữ nào dùng để diễn tả giải thuật O(n): (A) Constant (B) Cubic (C) Linear (D) Logarithmic

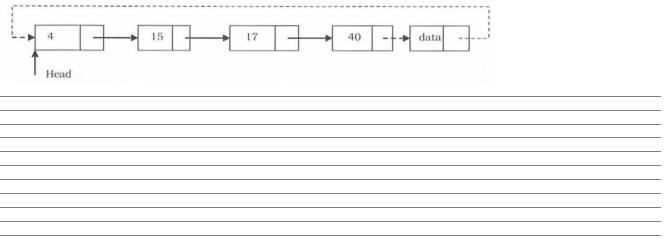
II. ĐIỀN KẾT QUẢ (4 điểm)

Câu 14.	Cho function sau:					
	<pre>int mystery_1(Node *tree = this.root): if(tree == nullptr):</pre>					
	$\begin{array}{ll} \textbf{return} & 0; \\ \textbf{elif} & (\texttt{tree.getLeft}() == \texttt{nullptr} \ \textbf{and} \ \texttt{tree.getRight}() == \texttt{nullptr}): \end{array}$					
	<pre>return 0; return 1 + mystery_1(tree.getLeft()) + mystery_1(tree.getRight());</pre>					
	Hãy cho biết công dụng của function này:					
Câu 15.	Chèn các khóa 12, 18, 13, 2, 3, 23, 5 và 15 vào 1 empty hash table có size là 10, sử dụng $h(k) = k \mod 10$ và linear probling. Hãy vẽ hash table kết quả (index từ 0).					
Câu 16.	<pre>Cho function sau: int mystery_2(Node *tree = this.root): if(tree == nullptr) return 0; else if (tree.getLeft() == nullptr and tree.getRight() == nullptr):</pre>					
	<pre>return 1; return 1 + mystery_2(tree.getLeft()) + mystery_2(tree.getRight());</pre>					
	Hãy cho biết công dụng của function này:					
Câu 18.	Cho array $A = [21, 1, 26, 45, 29, 28, 2, 9, 16, 49, 39, 27, 43, 34, 46, 40]$, lần lượt insert các phần tử của A theo thư tự index từ $0->15$ vào 1 cây empty AVL , hãy vẽ cây AVL cuối cùng được tạo ra.					
Câu 19.	Xóa lần lượt 9, và 29 của cây AVL ở câu trên. Hãy vẽ cây AVL sau khi xóa.					

Câu 20. Cho hình bên dưới:



Hãy viết các câu lệnh cần thiết để có thể chèn phần tử New node vào vị trí như hình sau, giả sử đã có 2 con trỏ Head và New node như hình:



Câu 21. Cho array A = [21, 1, 26, 45, 29, 28, 2, 9, 16, 49, 39, 27, 43, 34, 46, 40], lần lượt insert các phần tử của A theo thứ tự index từ 0->15 vào 1 empty BST, hãy vẽ BST cuối cùng được tạo ra.

·	

Câu 22. Cho array A = [54, 26, 93, 17, 77, 31, 44, 55, 20], lần lượt insert các phần tử của A theo thứ tự index từ 0->7 vào 1 empty MaxHeap, hãy vẽ cây Heap cuối cùng được tạo ra.

Câu 23. Xóa lần lượt phần tử lớn nhất, lớn nhì, và lớn thứ 3 của MaxHeap ở câu trên. Hãy vẽ MaxHeap sau khi xóa.

Câu 24. Cho array A = [54, 26, 93, 17, 77, 31, 44, 55, 20], sử dụng insertion sort sắp xếp A giảm dần, sau A bước sắp xếp (four complete passes) thì A = ?.

III. LẬP TRÌNH (5 điểm)

Câu 25. (1d) (CODING STYLE)

Sinh viên sẽ có điểm câu này nếu làm ít nhất 1 câu trong các câu tiếp theo và có coding style tốt. Điểm được chấm theo 3 mức: 0, 0.5, hoặc 1.

- Câu 26. Viết function bool moveK2first(Node* head, int k) để di chuyển node ở vị trí thứ k ra đầu danh sách. Giả sử phần tử đầu tiên nằm ở vị trí 1.
- Câu 27. Viết function int countLeave(TreeNode* root) để đếm số lá có trong 1 cây nhị phân tổng quát.
- Câu 28. Cho các khai báo sau:

```
struct StudentNode {
    int id;
    string name;
    StudentNode* next;
};

struct ClassNode {
    int id;
    string name;
    StudentNode* students;
    ClassNode* next;
};
```

Trong đó, cấu trúc StudentNode dùng để lưu trữ một node trong danh sách liên kết đơn chứa thông tin sinh viên của một lớp, ClassNode dùng để lưu trữ một node trong danh sách liên kết đơn chứa thông tin của các lớp đang quản lý trong một trường học. Thành phần students của ClassNode đại diện cho danh sách sinh viên của lớp đó.

Hãy định nghĩa hàm bool addStudent(ClassNode* list, int classId, int studentId, string studentName), trong đó list là con trỏ trỏ đến phần tử đầu tiên trong danh sách lớp, classId, studentId, studentName lần lượt là mã số lớp, mã số sinh viên và tên của sinh viên. Hàm thực thực hiện chức năng sau:

- Thêm sinh viên vào đúng lớp có classId tương ứng và trả về true. Nếu không tồn tại classId đó, không thực hiện và trả về false.
- Nếu tồn tại studentId (trong toàn bộ các lớp) không thực hiện và trả về false.
- Câu 29. Viết function void printSmallerNearestNode(BSTNode* root, int key) để tìm kiếm và in ra màn hình node duy nhất có giá trị nhỏ (hoặc bằng) hơn và gần nhất với giá trị key trong 1 BST (giá trị key vì thế có thể không tồn tại trong cây).

Chủ nhiệm Khoa/Bộ môn	Giảng viên ra đề
	ThS. Vương Bá Thịnh

Ho tên: MSSV: Trang 4/6 – Mã đề 5

 \square Không được phép dùng tài liệu

Câu 1. Giả sử cần chuyển biểu thức trung tố sang hậu tố,

cấu trúc dữ liệu nào phù hợp nhất được sử dụng

Đề thi số: 5

Câu 2.

Câu 3.

Câu 4.

Câu 5.

Câu 6.

để giải quyết bài toán này:

Môn thi: Cấu trúc dữ liệu và giải thuật

KIỂM TRA CUỐI KỲ (25/07/2020)

Thời gian làm bài: 120 phút

Câu 7. 1 hash table có length 10 sử dụng hash function

 $h(k)=k \ \mathrm{mod}\ 10,$ và linear probling. Sau khi chèn

6 giá trị vào 1 empty hash table, thì table chứa giá

⊠ Được phép dùng tài liệu giấy

I. TRẮC NGHIỆM (3 điểm)

A Queue C List	B Stack D Tree	(index của table đ	_, 42, 23, 34, 52, 46, 33,,]
àu 2. Xét đoạn code sau, đ		(A) 46, 42, 34, 52, 23, 33 (C) 34, 42, 23, 52, 33, 46	
$egin{array}{ll} {f int} & { m test} = 0 , { m i} , \\ {f for} & ({ m i} = 0; { m i} < 0; \\ {f for} & ({ m j} = 0; \\ { m test} = 0; \end{array}$		của 1 chuỗi các d tử trong stack tại	g stack để xác định sự cân bằng lấu đóng, mở ngoặc. Số phần một thời điểm (AT ANY ONE bao nhiêu khi phân tích chuỗi
\mathbf{C} $\mathrm{O}(n^2)$	(D) $O(\log n)$ nhất, giải thuật sort nào hiệu	(()(())(())) A 5 hoặc nhiều hơn C 4 E 3	B 1 D 2
A Insertion sort C Quick sort	B Selection sort D Merge sort	3 11 10 7 30 40,	duyệt NLR cho kết quả 14 2 1 duyệt LRN cho kết quả 1 3 2 7 luyệt LNR sẽ có kết quả là:
 àu 4. Dạng nào của List phì tử thứ i của list chứa A Lists hiện thực bằng 1 B Singly-linked lists hoặc phù hợp 	array	 A 1 2 3 7 10 11 14 30 4 B 14 1 2 3 10 11 7 40 3 C 1 2 3 14 7 10 11 40 3 D 1 2 3 7 10 14 30 11 4 	30 30
C Singly-linked lists D Doubly-linked lists (dar	nh sách liên kết 2 chiều)		của 1 full binary tree có depth $\cot x$ xem như có depth $= 0$) là:
	m front, rear) nào sẽ bị thay nử vào 1 queue đang có 1 phần	(C) 8 (E) 15	B 3D 5
A front		Câu 11. Chọn phát biểu sai	
từng đôi một. Giả sử	D cả 2 ố nguyên có giá trị khác nhau ở muốn tìm giá trị nhỏ nhất g hợp xấu nhất, giải pháp nào	phức tạp O(log n)	
A Sort A tăng dần, min l	à phần tử đầu tiên nin được chứa trong root	Câu 12. Trong trường hợp t hiệu quả nhất:	ốt nhất, giải thuật sort nào kém
A) Sort A tăng dần, min l B) Tạo 1 MinHeap từ A, r C Lặp qua cả dãy tìm ph D) Tạo 1 BST từ A, tìm p	ần tử min	(A) Insertion sort (C) Quick sort	B Selection sort D Bubble sort
Họ tên:	MSSV:		Trang $1/6~-$ Mã đề 5

\sim	Constant B Cubic Linear D Logarithmic
	II. ĐIỀN KẾT QUẢ (4 điểm)
C ≘ 1.4	
Cau 14.	<pre>Cho function sau: int mystery_1(Node *tree = this.root): if(tree == nullptr): return 0; elif (tree.getLeft() == nullptr and tree.getRight() == nullptr): return 0; return 1 + mystery_1(tree.getLeft()) + mystery_1(tree.getRight());</pre>
	Hãy cho biết công dụng của function này:
Câu 15.	Chèn các khóa 12, 18, 13, 2, 3, 23, 5 và 15 vào 1 empty hash table có size là 10, sử dụng $h(k) = k \mod 10$ và linear probling. Hãy vẽ hash table kết quả (index từ 0).
Câu 16.	Cho function sau:
	<pre>int mystery_2(Node *tree = this.root): if(tree == nullptr) return 0; else if (tree.getLeft() == nullptr and tree.getRight() == nullptr): return 1; return 1 + mystery_2(tree.getLeft()) + mystery_2(tree.getRight());</pre>
	Hãy cho biết công dụng của function này:
Câu 17.	Cho array $A = [54, 26, 93, 17, 77, 31, 44, 55, 20]$, sử dụng bubble sort sắp xếp A tăng dần , sau 4 bước sắp xếp (four complete passes) thì $A = ?$.
Câu 18.	Cho array $A = [21, 1, 26, 45, 29, 28, 2, 9, 16, 49, 39, 27, 43, 34, 46, 40]$, lần lượt insert các phần tử của A theo thứ tự index từ $0->15$ vào 1 cây empty AVL , hãy vẽ cây AVL cuối cùng được tạo ra.
Cân 10	Xóa lần lượt 9, và 29 của cây AVL ở câu trên. Hãy vẽ cây AVL sau khi xóa.

 ${\bf Câu}$ 13. Thuật ngữ nào dùng để diễn tả giải thuật ${\rm O(n)}\colon$

Họ tên:

MSSV:

Trang 2/6 — Mã đề 5

Câu 20.	Cho hình bên dưới:
	Head New node
	Hãy viết các câu lệnh cần thiết để có thể chèn phần tử New node vào vị trí như hình sau, giả sử đã có 2 con tr Head và New node như hình:
Câu 21.	Cho array $A = [21, 1, 26, 45, 29, 28, 2, 9, 16, 49, 39, 27, 43, 34, 46, 40], lần lượt insert các phần tử của A theo thư index từ 0->15 vào 1 empty BST, hãy vẽ BST cuối cùng được tạo ra.$
Câu 22.	Cho array $A = [54, 26, 93, 17, 77, 31, 44, 55, 20]$, lần lượt insert các phần tử của A theo thứ tự index từ $0->7$ vào empty MaxHeap, hãy vẽ cây Heap cuối cùng được tạo ra.

Câu 23.	Xóa lần lượt phần tử lớn nhất, lớn nhì, và lớn thứ 3 của MaxHeap ở câu trên. Hãy vẽ MaxHeap sau khi xóa.
Câu 24.	Cho array $A = [54, 26, 93, 17, 77, 31, 44, 55, 20]$, sử dụng insertion sort sắp xếp A giảm dần , sau 4 bước sắp xếp (four complete passes) thì $A = ?$.

III. LẬP TRÌNH (5 điểm)

Câu 25. (1d) (CODING STYLE)

Sinh viên sẽ có điểm câu này nếu làm ít nhất 1 câu trong các câu tiếp theo và có coding style tốt. Điểm được chấm theo 3 mức: 0, 0.5, hoặc 1.

- Câu 26. Viết function bool moveK2first(Node* head, int k) để di chuyển node ở vị trí thứ k ra đầu danh sách. Giả sử phần tử đầu tiên nằm ở vị trí 1.
- Câu 27. Viết function int countLeave(TreeNode* root) để đếm số lá có trong 1 cây nhị phân tổng quát.
- Câu 28. Cho các khai báo sau:

```
struct StudentNode {
    int id;
    string name;
    StudentNode* next;
};

struct ClassNode {
    int id;
    string name;
    StudentNode* students;
    ClassNode* next;
};
```

Trong đó, cấu trúc StudentNode dùng để lưu trữ một node trong danh sách liên kết đơn chứa thông tin sinh viên của một lớp, ClassNode dùng để lưu trữ một node trong danh sách liên kết đơn chứa thông tin của các lớp đang quản lý trong một trường học. Thành phần students của ClassNode đại diện cho danh sách sinh viên của lớp đó.

Hãy định nghĩa hàm **bool addStudent(ClassNode* list, int classId, int studentId, string studentName)**, trong đó list là con trỏ trỏ đến phần tử đầu tiên trong danh sách lớp, classId, studentId, studentName lần lượt là mã số lớp, mã số sinh viên và tên của sinh viên. Hàm thực thực hiện chức năng sau:

- Thêm sinh viên vào đúng lớp có class Id tương ứng và trả về true. Nếu không tồn tại class Id đó, không thực hiện và trả về false.
- Nếu tồn tại studentId (trong toàn bộ các lớp) không thực hiện và trả về false.
- Câu 29. Viết function void printSmallerNearestNode(BSTNode* root, int key) để tìm kiếm và in ra màn hình node duy nhất có giá trị nhỏ (hoặc bằng) hơn và gần nhất với giá trị key trong 1 BST (giá trị key vì thế có thể không tồn tại trong cây).

Ho tên: MSSV: Trang 4/6 – Mã đề 5

Đề thi số: 7

Môn thi: Cấu trúc dữ liêu và giải thuật

☐ Không được phép dùng tài liệu

Họ và tên:	
MSSV:	

Thời gian làm bài: 120 phút

☑ Được phép dùng tài liệu giấy

Đề thi gồm 6 trang.

KIỂM TRA CUỐI KỲ (25/07/2020)

Chú ý: Sinh viên làm trực tiếp trên đề thi; môi câu hỏi trắc nghiệm chỉ chọn một đáp án đúng nhất. Các khai báo vê tl viện, các hàm cơ bản và khai báo khác (nếu cần thiết) được xem như đầy đủ		
	I. TRẮC	NGHIỆM (3 điểm)
<pre>int test for (i = for (</pre>	de sau, độ phức tạp Big-O: = 0, i, j; 0; i < n; i++): j = 0; j < i; j++): est = test + i * j;	Câu 8. Giả sử cần chuyển biểu thức trung tố sang hậu tố cấu trúc dữ liệu nào phù hợp nhất được sử dụng để giải quyết bài toán này: (A) Queue (B) Tree (C) List (D) Stack
$ \begin{array}{c} $	(B) O(log n) (D) O(1)	Câu 9. 1 hash table có length 10 sử dụng hash function $h(k) = k \mod 10$, và linear probling. Sau khi chèr

Câu 2. Xét giải thuật dùng stack để xác định sự cân bằng của 1 chuỗi các dấu đóng, mở ngoặc. Số phần tử trong stack tại một thời điểm (AT ANY ONE TIME) lớn nhất là bao nhiêu khi phân tích chuỗi sau: (()(())(())(()))

- 5 hoặc nhiều hơn

Câu 3. Chọn phát biểu sai

- (A) Linear Searching có độ phức tạp O(n)
- (B) Searching trên cây AVL có độ phức tạp O(ln n)
- Binary Searching trên Ordered Array có đô phức tạp O(log n)
- (D) Searching trên Heap có độ phức tạp O(log n)

Câu 4. Trong trường hợp tốt nhất, giải thuật sort nào hiệu quả nhất:

- (A) Insertion sort
- (B) Merge sort
- (C) Quick sort
- (D) Selection sort

Câu 5. Số node tối thiểu của 1 full binary tree có depth = 3 (cây chỉ có root xem như có depth = 0) là:

Câu 6. Thuật ngữ nào dùng để diễn tả giải thuật O(n):

- (A) Constant
- (B) Logarithmic
- (C) Linear
- Cubic

Câu 7. Trong trường hợp tốt nhất, giải thuật sort nào kém hiệu quả nhất:

- Insertion sort
- (B) Bubble sort
- Quick sort
- (D) Selection sort

 $h(k) = k \mod 10$, và linear probling. Sau khi chèn 6 giá trị vào 1 empty hash table, thì table chứa giá trị như sau: [_, _, 42, 23, 34, 52, 46, 33, _, _] (index của table được đánh từ 0). Các key values được chèn theo thứ tự trước sau như thế nào?

- 46, 42, 34, 52, 23, 33
- **(B)** 46, 34, 42, 23, 52, 33
- 34, 42, 23, 52, 33, 46
- (D) 42, 46, 33, 23, 34, 52

Câu 10. Con trỏ (pointers gồm front, rear) nào sẽ bị thay đổi khi insert 1 phần tử vào 1 queue đang có 1 phần tử:

- front
- rear
- không có con trỏ nào thay đổi

Câu 11. Cho array A chứa n số nguyên có giá trị khác nhau từng đôi một. Giả sử muốn tìm giá trị nhỏ nhất trong A, trong trường hợp xấu nhất, giải pháp nào sau đây hiệu quả nhất:

- Sort A tăng dần, min là phần tử đầu tiên
- (B) Tao 1 BST từ A, tìm phần tử min
- (C) Lặp qua cả dãy tìm phần tử min
- Tạo 1 MinHeap từ A, min được chứa trong root

Câu 12. Dạng nào của List phù hợp để trả lời câu hỏi "Phần tử thứ i của list chứa gì?":

- (A) Lists hiện thực bằng 1 array
- (B) Doubly-linked lists (danh sách liên kết 2 chiều)
- (C) Singly-linked lists
- Singly-linked lists hoăc Doubly-linked lists đều phù hợp

Câu 13. 1 cây nhi phân khi duyệt NLR cho kết quả 14 2 1 3 11 10 7 30 40, duyệt LRN cho kết quả 1 3 2 7 ${\bf 10}~{\bf 40}~{\bf 30}~{\bf 11}~{\bf 14},$ duyệt LNR sẽ có kết quả là:

- (A) 1 2 3 7 10 11 14 30 40 (B) 1 2 3 7 10 14 30 11 40
- (C) 1 2 3 14 7 10 11 40 30 (D) 14 1 2 3 10 11 7 40 30

II. ĐIỆN KẾT QUẢ (4 điểm)

Câu 14. Cho function sau:

```
int mystery_2(Node *tree = this.root):
    if(tree == nullptr)
        return 0;
    else if (tree.getLeft() == nullptr and tree.getRight() == nullptr):
        return 1;
    return 1 + mystery_2(tree.getLeft()) + mystery_2(tree.getRight());
```

Hãy cho biết công dụng của function này:

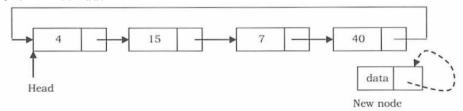
Câu 15. Cho array A = [54, 26, 93, 17, 77, 31, 44, 55, 20], sử dụng bubble sort sắp xếp A tăng dần, sau A bước sắp xếp (four complete passes) thì A = ?.

Câu 16. Cho function sau:

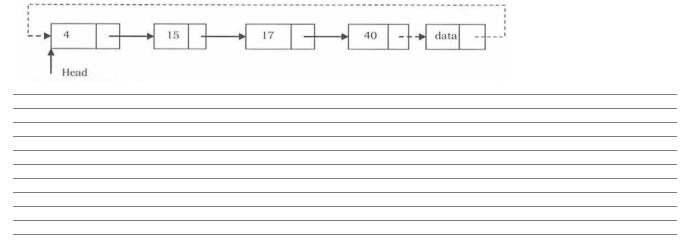
```
int mystery_1(Node *tree = this.root):
    if(tree == nullptr):
        return 0;
    elif (tree.getLeft() == nullptr and tree.getRight() == nullptr):
        return 0;
    return 1 + mystery_1(tree.getLeft()) + mystery 1(tree.getRight());
```

Hãy cho biết công dung của function này:

Câu 17. Cho hình bên dưới:



Hãy viết các câu lệnh cần thiết để có thể chèn phần tử New node vào vị trí như hình sau, giả sử đã có 2 con trỏ Head và New node như hình:



Câu 18. Chèn các khóa 12, 18, 13, 2, 3, 23, 5 và 15 vào 1 empty hash table có size là 10, sử dụng $h(k) = k \mod 10$ và linear probling. Hãy vẽ hash table kết quả (index từ 0).

Câu 19. Cho array A = [54, 26, 93, 17, 77, 31, 44, 55, 20], sử dụng insertion sort sắp xếp A giảm dần, sau A bước sắp xếp (four complete passes) thì A = ?.

Ho tên:

âu 20.	Cho array $A = [54, 26, 93, 17, 77, 31, 44, 55, 20]$, lần lượt insert các phần tử của A theo thứ tự index từ 0->7 vàc empty MaxHeap, hãy vẽ cây Heap cuối cùng được tạo ra.
âu 2 1.	Xóa lần lượt phần tử lớn nhất, lớn nhì, và lớn thứ 3 của MaxHeap ở câu trên. Hãy vẽ MaxHeap sau khi xóa.
	Cho array $A = [21, 1, 26, 45, 29, 28, 2, 9, 16, 49, 39, 27, 43, 34, 46, 40]$, lần lượt insert các phần tử của A theo t tự index từ $0->15$ vào 1 cây empty AVL , hãy vẽ cây AVL cuối cùng được tạo ra.
âu 23.	Xóa lần lượt 9, và 29 của cây AVL ở câu trên. Hãy vẽ cây AVL sau khi xóa.

tự index từ 0->15 vào 1 empty BST, hãy vẽ BST cuối cùng được tạo ra.

-

III. LẬP TRÌNH (5 điểm)

Câu 25. (1d) (CODING STYLE)

Sinh viên sẽ có điểm câu này nếu làm ít nhất 1 câu trong các câu tiếp theo và có coding style tốt. Điểm được chấm theo 3 mức: 0, 0.5, hoặc 1.

- Câu 26. Viết function bool moveK2first(Node* head, int k) để di chuyển node ở vị trí thứ k ra đầu danh sách. Giả sử phần tử đầu tiên nằm ở vị trí 1.
- Câu 27. Viết function int countLeave(TreeNode* root) để đếm số lá có trong 1 cây nhị phân tổng quát.
- Câu 28. Cho các khai báo sau:

```
struct StudentNode {
    int id;
    string name;
    StudentNode* next;
};

struct ClassNode {
    int id;
    string name;
    StudentNode* students;
    ClassNode* next;
};
```

Trong đó, cấu trúc StudentNode dùng để lưu trữ một node trong danh sách liên kết đơn chứa thông tin sinh viên của một lớp, ClassNode dùng để lưu trữ một node trong danh sách liên kết đơn chứa thông tin của các lớp đang quản lý trong một trường học. Thành phần students của ClassNode đại diện cho danh sách sinh viên của lớp đó.

Hãy định nghĩa hàm **bool addStudent(ClassNode* list, int classId, int studentId, string studentName)**, trong đó list là con trỏ trỏ đến phần tử đầu tiên trong danh sách lớp, classId, studentId, studentName lần lượt là mã số lớp, mã số sinh viên và tên của sinh viên. Hàm thực thực hiện chức năng sau:

- Thêm sinh viên vào đúng lớp có classId tương ứng và trả về true. Nếu không tồn tại classId đó, không thực hiện và trả về false.
- Nếu tồn tại studentId (trong toàn bộ các lớp) không thực hiện và trả về false.
- Câu 29. Viết function void printSmallerNearestNode(BSTNode* root, int key) để tìm kiếm và in ra màn hình node duy nhất có giá trị nhỏ (hoặc bằng) hơn và gần nhất với giá trị key trong 1 BST (giá trị key vì thế có thể không tồn tại trong cây).

Chủ nhiệm Khoa/Bộ môn	Giảng viên ra đề
	ThS. Vương Bá Thịnh

Ho tên: MSSV: Trang 4/6 – Mã đề 7

Môn thi: Cấu trúc dữ liệu và giải thuật

KIỂM TRA CUỐI KỲ (25/07/2020)

Thời gian làm bài: 120 phút

\Box Không được phép dùng tài liệu $\hfill \boxtimes$ Được Đề thi số: 7	ợc phép dùng tài liệu giấy
I. TRẮC NGI	HIỆM (3 điểm)
Câu 1. Xét đoạn code sau, độ phức tạp Big-O: int test = 0, i, j; for (i = 0; i < n; i++): for (j = 0; j < i; j++): test = test + i * j;	Câu 8. Giả sử cần chuyển biểu thức trung tố sang hậu tố cấu trúc dữ liệu nào phù hợp nhất được sử dụng để giải quyết bài toán này: (A) Queue (B) Tree (C) List (D) Stack
A O(n) B O(log n) C O(n²) D O(1) Câu 2. Xét giải thuật dùng stack để xác định sự cân bằng của 1 chuỗi các dấu đóng, mở ngoặc. Số phần tử trong stack tại một thời điểm (AT ANY ONE TIME) lớn nhất là bao nhiêu khi phân tích chuỗi sau: (()(())(())) A 5 hoặc nhiều hơn B 1 C 2 D 4	 Câu 9. 1 hash table có length 10 sử dụng hash function h(k) = k mod 10, và linear probling. Sau khi chèr 6 giá trị vào 1 empty hash table, thì table chứa giá trị như sau: [_,_,_, 42, 23, 34, 52, 46, 33,,_ (index của table được đánh từ 0). Các key values được chèn theo thứ tự trước sau như thế nào? (A) 46, 42, 34, 52, 23, 33 B 46, 34, 42, 23, 52, 33 (B) 34, 42, 23, 52, 33, 46 D 42, 46, 33, 23, 34, 52 (Câu 10. Con trỏ (pointers gồm front, rear) nào sẽ bị thay đổi khi insert 1 phần tử vào 1 queue đang có 1 phần tử:
 Câu 3. Chọn phát biểu sai A Linear Searching có độ phức tạp O(n) B Searching trên cây AVL có độ phức tạp O(ln n) C Binary Searching trên Ordered Array có độ phức tạp O(log n) D Searching trên Heap có độ phức tạp O(log n) 	 A front C rear D không có con trỏ nào thay đổi Câu 11. Cho array A chứa n số nguyên có giá trị khác nhau từng đôi một. Giả sử muốn tìm giá trị nhỏ nhất trong A, trong trường hợp xấu nhất, giải pháp nào sau đây hiệu quả nhất:
Câu 4. Trong trường hợp tốt nhất, giải thuật sort nào hiệu quả nhất: A Insertion sort C Quick sort D Selection sort	 A Sort A tăng dần, min là phần tử đầu tiên B Tạo 1 BST từ A, tìm phần tử min C Lặp qua cả dãy tìm phần tử min D Tạo 1 MinHeap từ A, min được chứa trong root
Câu 5. Số node tối thiểu của 1 full binary tree có depth = 3 (cây chỉ có root xem như có depth = 0) là: A 7 B 3 C 5 E 15	 Câu 12. Dạng nào của List phù hợp để trả lời câu hỏi "Phần tử thứ i của list chứa gì?": A Lists hiện thực bằng 1 array B Doubly-linked lists (danh sách liên kết 2 chiều) C Singly-linked lists D Singly-linked lists hoặc Doubly-linked lists đều
Câu 6. Thuật ngữ nào dùng để diễn tả giải thuật O(n): (A) Constant (B) Logarithmic (C) Linear (D) Cubic	phù hợp Câu 13. 1 cây nhị phân khi duyệt NLR cho kết quả 14 2 1 3 11 10 7 30 40, duyệt LRN cho kết quả 1 3 2 7 10 40 30 11 14, duyệt LNR sẽ có kết quả là:
 Câu 7. Trong trường hợp tốt nhất, giải thuật sort nào kém hiệu quả nhất: (A) Insertion sort (B) Bubble sort (C) Quick sort (D) Selection sort 	(A) 1 2 3 7 10 11 14 30 40 (B) 1 2 3 7 10 14 30 11 40 (C) 1 2 3 14 7 10 11 40 30 (D) 14 1 2 3 10 11 7 40 30

MSSV:

Họ tên:

Trang 1/6 – Mã đề 7

II. ĐIỀN KẾT QUẢ (4 điểm)

Câu 14. Cho function sau:

```
int mystery_2(Node *tree = this.root):
    if(tree == nullptr)
        return 0;
    else if (tree.getLeft() == nullptr and tree.getRight() == nullptr):
        return 1;
    return 1 + mystery_2(tree.getLeft()) + mystery_2(tree.getRight());
```

Hãy cho biết công dụng của function này:

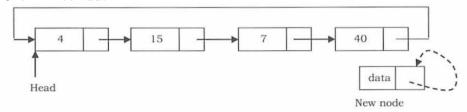
Câu 15. Cho array A = [54, 26, 93, 17, 77, 31, 44, 55, 20], sử dụng bubble sort sắp xếp A tăng dần, sau A bước sắp xếp (four complete passes) thì A = ?.

Câu 16. Cho function sau:

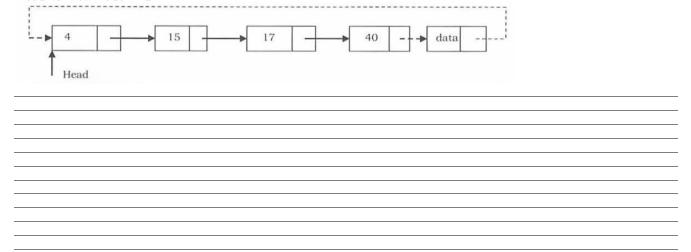
```
int mystery_1(Node *tree = this.root):
    if(tree == nullptr):
        return 0;
    elif (tree.getLeft() == nullptr and tree.getRight() == nullptr):
        return 0;
    return 1 + mystery 1(tree.getLeft()) + mystery 1(tree.getRight());
```

Hãy cho biết công dụng của function này:

Câu 17. Cho hình bên dưới:



Hãy viết các câu lệnh cần thiết để có thể chèn phần tử New node vào vị trí như hình sau, giả sử đã có 2 con trỏ Head và New node như hình:



Câu 18. Chèn các khóa 12, 18, 13, 2, 3, 23, 5 và 15 vào 1 empty hash table có size là 10, sử dụng $h(k) = k \mod 10$ và linear probling. Hãy vẽ hash table kết quả (index từ 0).

Ho tên:

Câu 19.	Cho array $A = [54, 26, 93, 17, 77, 31, 44, 55, 20]$, sử dụng insertion sort sắp xếp A giảm dần , sau 4 bước sắp xếp (four complete passes) thì $A = ?$.
Câu 20.	Cho array $A = [54, 26, 93, 17, 77, 31, 44, 55, 20]$, lần lượt insert các phần tử của A theo thứ tự index từ 0->7 vào 1 empty MaxHeap, hãy vẽ cây Heap cuối cùng được tạo ra.
Câu 21.	Xóa lần lượt phần tử lớn nhất, lớn nhì, và lớn thứ 3 của MaxHeap ở câu trên. Hãy vẽ MaxHeap sau khi xóa.
Câu 22.	Cho array $A = [21, 1, 26, 45, 29, 28, 2, 9, 16, 49, 39, 27, 43, 34, 46, 40]$, lần lượt insert các phần tử của A theo thứ tự index từ $0->15$ vào 1 cây empty AVL, hãy vẽ cây AVL cuối cùng được tạo ra.
Câu 23.	Xóa lần lượt 9, và 29 của cây AVL ở câu trên. Hãy vẽ cây AVL sau khi xóa.

-			
-			
-			

III. LẬP TRÌNH (5 điểm)

Câu 25. (1d) (CODING STYLE)

Sinh viên sẽ có điểm câu này nếu làm ít nhất 1 câu trong các câu tiếp theo và có coding style tốt. Điểm được chấm theo 3 mức: 0, 0.5, hoặc 1.

- Câu 26. Viết function bool moveK2first(Node* head, int k) để di chuyển node ở vị trí thứ k ra đầu danh sách. Giả sử phần tử đầu tiên nằm ở vi trí 1.
- Câu 27. Viết function int countLeave(TreeNode* root) để đếm số lá có trong 1 cây nhị phân tổng quát.
- Câu 28. Cho các khai báo sau:

```
struct StudentNode {
    int id;
    string name;
    StudentNode* next;
};

struct ClassNode {
    int id;
    string name;
    StudentNode* students;
    ClassNode* next;
};
```

Trong đó, cấu trúc StudentNode dùng để lưu trữ một node trong danh sách liên kết đơn chứa thông tin sinh viên của một lớp, ClassNode dùng để lưu trữ một node trong danh sách liên kết đơn chứa thông tin của các lớp đang quản lý trong một trường học. Thành phần students của ClassNode đại diện cho danh sách sinh viên của lớp đó.

Hãy định nghĩa hàm bool addStudent(ClassNode* list, int classId, int studentId, string studentName), trong đó list là con trỏ trỏ đến phần tử đầu tiên trong danh sách lớp, classId, studentId, studentName lần lượt là mã số lớp, mã số sinh viên và tên của sinh viên. Hàm thực thực hiện chức năng sau:

- Thêm sinh viên vào đúng lớp có class Id tương ứng và trả về true. Nếu không tồn tại class Id đó, không thực hiện và trả về false.
- Nếu tồn tại studentId (trong toàn bộ các lớp) không thực hiện và trả về false.
- Câu 29. Viết function void printSmallerNearestNode(BSTNode* root, int key) để tìm kiếm và in ra màn hình node duy nhất có giá trị nhỏ (hoặc bằng) hơn và gần nhất với giá trị key trong 1 BST (giá trị key vì thế có thể không tồn tại trong cây).

Ho tên: MSSV: Trang 4/6 – Mã đề 7

TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH

Họ và tên:	
MSSV:	

$ext{KI\r{E}M}$ TRA CUỐI KỲ (25/07/2020)

KIEW TITA COOL	(20/01/2020)
Môn thi: Cấu trúc dữ liệu và giải thuậ	t Thời gian làm bài: 120 phút
Đề thi số: 9 Đề Chú ý: Sinh viên làm trực tiếp trên đề thi; mỗi câu hỏi trắc	⊠ Được phép dùng tài liệu giấy ề thi gồm 6 trang. nghiệm chỉ chọn một đáp án đúng nhất. Các khai báo về thư c (nếu cần thiết) được xem như đầy đủ
I. TRẮC NGI	$\mathrm{HI}\mathrm{\hat{E}M}$ (3 điểm)
 Câu 1. Giả sử cần chuyển biểu thức trung tố sang hậu tố, cấu trúc dữ liệu nào phù hợp nhất được sử dụng để giải quyết bài toán này: (A) Stack (B) Queue (C) Tree (D) List Câu 2. Trong trường hợp tốt nhất, giải thuật sort nào hiệu 	Câu 7. Trong trường hợp tốt nhất, giải thuật sort nào kém hiệu quả nhất: (A) Selection sort (B) Insertion sort (C) Bubble sort (D) Quick sort (Câu 8. Xét đoạn code sau, độ phức tạp Big-O: int test = 0, i, j;
quả nhất: (A) Selection sort (C) Merge sort (D) Quick sort	for $(i = 0; i < n; i++):$ for $(j = 0; j < i; j++):$ test = test + i * j;
Câu 3. Con trỏ (pointers gồm front, rear) nào sẽ bị thay đổi khi insert 1 phần tử vào 1 queue đang có 1 phần tử: (A) không có con trỏ nào thay đổi (B) front (C) cả 2 (D) rear	(A) O(1) (B) O(n) (C) O(log n) (D) O(n ²) Câu 9. Số node tối thiểu của 1 full binary tree có depth = 3 (cây chỉ có root xem như có depth = 0) là: (A) 3 (B) 7 (C) 15 (D) 5 (E) 8
 Câu 4. Dạng nào của List phù hợp để trả lời câu hỏi "Phần tử thứ i của list chứa gì?": A Singly-linked lists hoặc Doubly-linked lists đều phù hợp B Lists hiện thực bằng 1 array C Doubly-linked lists (danh sách liên kết 2 chiều) D Singly-linked lists 	Câu 10. Chọn phát biểu sai A Searching trên Heap có độ phức tạp O(log n) B Linear Searching có độ phức tạp O(n) C Searching trên cây AVL có độ phức tạp O(ln n) Binary Searching trên Ordered Array có độ phức tạp O(log n)
 Câu 5. 1 hash table có length 10 sử dụng hash function h(k) = k mod 10, và linear probling. Sau khi chèn 6 giá trị vào 1 empty hash table, thì table chứa giá trị như sau: [_,_, 42, 23, 34, 52, 46, 33, _, _] (index của table được đánh từ 0). Các key values được chèn theo thứ tự trước sau như thế nào? (A) 42, 46, 33, 23, 34, 52 (B) 46, 42, 34, 52, 23, 33 (C) 46, 34, 42, 23, 52, 33 (D) 34, 42, 23, 52, 33, 46 	Câu 11. Thuật ngữ nào dùng để diễn tả giải thuật O(n): (A) Cubic (B) Constant (C) Logarithmic (D) Linear Câu 12. Cho array A chứa n số nguyên có giá trị khác nhau từng đôi một. Giả sử muốn tìm giá trị nhỏ nhất trong A, trong trường hợp xấu nhất, giải pháp nào sau đây hiệu quả nhất: (A) Tạo 1 MinHeap từ A, min được chứa trong root
Câu 6. Xét giải thuật dùng stack để xác định sự cân bằng của 1 chuỗi các dấu đóng, mở ngoặc. Số phần tử trong stack tại một thời điểm (AT ANY ONE TIME) lớn nhất là bao nhiêu khi phân tích chuỗi sau: (()(())(())) (A) 1 (B) 5 hoặc nhiều hơn (C) 3 (D) 2	(B) Sort A tăng dần, min là phần tử đầu tiên (C) Tạo 1 BST từ A, tìm phần tử min (D) Lặp qua cả dãy tìm phần tử min (Câu 13. 1 cây nhị phân khi duyệt NLR cho kết quả 14 2 1 3 11 10 7 30 40, duyệt LRN cho kết quả 1 3 2 7 10 40 30 11 14, duyệt LNR sẽ có kết quả là: (A) 14 1 2 3 10 11 7 40 30 (B) 1 2 3 7 10 11 14 30 40 (C) 1 2 3 7 10 14 30 11 40 (D) 1 2 3 14 7 10 11 40 30
II. ĐIEN KET	QUẢ (4 điểm)

Câu 14.	Cho function sau:			
	<pre>int mystery_1(Node *tree = this.root): if(tree == nullptr): return 0; elif (tree.getLeft() == nullptr and tree.getRight() == nullptr): return 0; return 1 + mystery_1(tree.getLeft()) + mystery_1(tree.getRight());</pre>			
	Hãy cho biết công dụng của function này:			
Câu 15.	Cho array $A = [54, 26, 93, 17, 77, 31, 44, 55, 20]$, sử dụng insertion sort sắp xếp A giảm dần , sau 4 bước sắp xếp (four complete passes) thì $A = ?$.			
Câu 16.	Cho function sau:			
	<pre>int mystery_2(Node *tree = this.root): if(tree == nullptr) return 0;</pre>			
	<pre>else if (tree.getLeft() == nullptr and tree.getRight() == nullptr): return 1;</pre>			
	return 1; return 1 + mystery_2(tree.getLeft()) + mystery_2(tree.getRight());			
	Hãy cho biết công dụng của function này:			
Câu 17.	Chèn các khóa 12, 18, 13, 2, 3, 23, 5 và 15 vào 1 empty hash table có size là 10, sử dụng $h(k) = k \mod 10$ và linear probling. Hãy vẽ hash table kết quả (index từ 0).			
Câu 18.	Cho array $A = [54, 26, 93, 17, 77, 31, 44, 55, 20]$, sử dụng bubble sort sắp xếp A tăng dần , sau 4 bước sắp xếp (four complete passes) thì $A = ?$.			
Câu 19.	Cho array $A = [54, 26, 93, 17, 77, 31, 44, 55, 20]$, lần lượt insert các phần tử của A theo thứ tự index từ $0->7$ vào 1 empty MaxHeap, hãy vẽ cây Heap cuối cùng được tạo ra.			
Câu 20.	Xóa lần lượt phần tử lớn nhất, lớn nhì, và lớn thứ 3 của MaxHeap ở câu trên. Hãy vẽ MaxHeap sau khi xóa.			

L. (Cho hình bên dưới:
	Head New node
I	Hãy viết các câu lệnh cần thiết để có thể chèn phần tử New node vào vị trí như hình sau, giả sử đã có 2 Head và New node như hình:
	Head 15 17 40 - data
_ _ _	
_	
- -	
	Cho array $A = [21, 1, 26, 45, 29, 28, 2, 9, 16, 49, 39, 27, 43, 34, 46, 40]$, lần lượt insert các phần tử của A tự index từ $0->15$ vào 1 cây empty AVL, hãy vẽ cây AVL cuối cùng được tạo ra.
-	
- -	
- -	
-	Ván làn hart 0 nà 20 nà nôn AVI à nôn tuôn Hữa nữ nôn AVI non libi ná
). 2 _ _ _	Xóa lần lượt 9, và 29 của cây AVL ở câu trên. Hãy vẽ cây AVL sau khi xóa.
- -	
-	
_	

-

III. LẬP TRÌNH (5 điểm)

Câu 25. (1d) (CODING STYLE)

Sinh viên sẽ có điểm câu này nếu làm ít nhất 1 câu trong các câu tiếp theo và có coding style tốt. Điểm được chấm theo 3 mức: 0, 0.5, hoặc 1.

- Câu 26. Viết function bool moveK2first(Node* head, int k) để di chuyển node ở vị trí thứ k ra đầu danh sách. Giả sử phần tử đầu tiên nằm ở vị trí 1.
- Câu 27. Viết function int countLeave(TreeNode* root) để đếm số lá có trong 1 cây nhị phân tổng quát.
- Câu 28. Cho các khai báo sau:

```
struct StudentNode {
    int id;
    string name;
    StudentNode* next;
};

struct ClassNode {
    int id;
    string name;
    StudentNode* students;
    ClassNode* next;
};
```

Trong đó, cấu trúc StudentNode dùng để lưu trữ một node trong danh sách liên kết đơn chứa thông tin sinh viên của một lớp, ClassNode dùng để lưu trữ một node trong danh sách liên kết đơn chứa thông tin của các lớp đang quản lý trong một trường học. Thành phần students của ClassNode đại diện cho danh sách sinh viên của lớp đó.

Hãy định nghĩa hàm **bool addStudent(ClassNode* list, int classId, int studentId, string studentName)**, trong đó list là con trỏ trỏ đến phần tử đầu tiên trong danh sách lớp, classId, studentId, studentName lần lượt là mã số lớp, mã số sinh viên và tên của sinh viên. Hàm thực thực hiện chức năng sau:

- Thêm sinh viên vào đúng lớp có class Id tương ứng và trả về true. Nếu không tồn tại class Id đó, không thực hiện và trả về false.
- Nếu tồn tại studentId (trong toàn bộ các lớp) không thực hiện và trả về false.
- Câu 29. Viết function void printSmallerNearestNode(BSTNode* root, int key) để tìm kiếm và in ra màn hình node duy nhất có giá trị nhỏ (hoặc bằng) hơn và gần nhất với giá trị key trong 1 BST (giá trị key vì thế có thể không tồn tại trong cây).

Chủ nhiệm Khoa/Bộ môn	Giảng viên ra đề
	ThS. Vương Bá Thịnh

Ho tên: MSSV: Trang 4/6 – Mã đề 9

 \square Không được phép dùng tài liệu

Câu 1. Giả sử cần chuyển biểu thức trung tố sang hậu tố,

cấu trúc dữ liệu nào phù hợp nhất được sử dụng

(B) Queue

List

Đề thi số: 9

A Stack

Tree

để giải quyết bài toán này:

Môn thi: Cấu trúc dữ liệu và giải thuật

KIỂM TRA CUỐI KỲ (25/07/2020)

Thời gian làm bài: 120 phút

hiệu quả nhất:

Selection sort

Bubble sort

Câu 7. Trong trường hợp tốt nhất, giải thuật sort nào kém

(B) Insertion sort

Quick sort

⊠ Được phép dùng tài liệu giấy

I. TRẮC NGHIỆM (3 điểm)

 Câu 2. Trong trường hợp tốt nhất, giải thuật sort nào hiệu quả nhất: A Selection sort C Merge sort D Quick sort 	Câu 8. Xét đoạn code sau, độ phức tạp Big-O: int test = 0, i, j; for (i = 0; i < n; i++): for (j = 0; j < i; j++): test = test + i * j;
Câu 3. Con trỏ (pointers gồm front, rear) nào sẽ bị thay đổi khi insert 1 phần tử vào 1 queue đang có 1 phần tử:	$ \begin{array}{c c} \textbf{(A)} & O(1) & \textbf{(B)} & O(n) \\ \hline \textbf{(C)} & O(\log n) & \textbf{(D)} & O(n^2) \\ \end{array} $
 A không có con trỏ nào thay đổi B front C cả 2 D rear Câu 4. Dạng nào của List phù hợp để trả lời câu hỏi "Phần tử thứ i của list chứa gì?": 	Câu 9. Số node tối thiểu của 1 full binary tree có depth = 3 (cây chỉ có root xem như có depth = 0) là: A 3 C 15 D 5
 A Singly-linked lists hoặc Doubly-linked lists đều phù hợp B Lists hiện thực bằng 1 array C Doubly-linked lists (danh sách liên kết 2 chiều) D Singly-linked lists 	Câu 10. Chọn phát biểu sai A Searching trên Heap có độ phức tạp O(log n) B Linear Searching có độ phức tạp O(n) C Searching trên cây AVL có độ phức tạp O(ln
 Câu 5. 1 hash table có length 10 sử dụng hash function h(k) = k mod 10, và linear probling. Sau khi chèn 6 giá trị vào 1 empty hash table, thì table chứa giá trị như sau: [_,_, 42, 23, 34, 52, 46, 33,, _] (index của table được đánh từ 0). Các key values được chèn theo thứ tự trước sau như thế nào? (A) 42, 46, 33, 23, 34, 52 (B) 46, 42, 34, 52, 23, 33 (C) 46, 34, 42, 23, 52, 33 (D) 34, 42, 23, 52, 33, 46 	n) Binary Searching trên Ordered Array có độ phức tạp O(log n) Câu 11. Thuật ngữ nào dùng để diễn tả giải thuật O(n): (A) Cubic (B) Constant (C) Logarithmic (D) Linear
Câu 6. Xét giải thuật dùng stack để xác định sự cân bằng của 1 chuỗi các dấu đóng, mở ngoặc. Số phần tử trong stack tại một thời điểm (AT ANY ONE TIME) lớn nhất là bao nhiều khi phân tích chuỗi sau: (()(())(())) (A) 1 (B) 5 hoặc nhiều hơn (C) 3 (D) 2	 Câu 12. Cho array A chứa n số nguyên có giá trị khác nhau từng đôi một. Giả sử muốn tìm giá trị nhỏ nhất trong A, trong trường hợp xấu nhất, giải pháp nào sau đây hiệu quả nhất: (A) Tạo 1 MinHeap từ A, min được chứa trong root (B) Sort A tăng dần, min là phần tử đầu tiên (C) Tạo 1 BST từ A, tìm phần tử min (D) Lặp qua cả dãy tìm phần tử min
Họ tên: MSSV:	Trang $1/6~-$ Mã đề 9

Câu 13.	1 cây nhị phân khi duyệt NLR cho kết quả 14 2 1 3 11 10 7 30 40 , duyệt LRN cho kết quả 1 3 2 7 10 40 30 11 14 , duyệt LNR sẽ có kết quả là:			
(A) 1	4 1 2 3 10 11 7 40 30			
B 1	2 3 7 10 11 14 30 40			
© 1	2 3 7 10 14 30 11 40			
$\boxed{\mathbf{D}}$ 1	2 3 14 7 10 11 40 30			
	II. ĐIỀN KẾT	${ m QU}{ m \AA}$ (4 điểm)		
Câu 14.	Cho function sau:			
	<pre>return 0; elif (tree.getLeft() == nullptr and tree.getRight() == nullptr): return 0;</pre>			
	<pre>return 1 + mystery_1(tree.getLeft()) + mystery_1(tree.getRight());</pre>			
Hãy cho biết công dụng của function này:				
Câu 15.	Cho array $A = [54, 26, 93, 17, 77, 31, 44, 55, 20]$, sử dụng insertion sort sắp xếp A giảm dần , sau 4 bước sắp xếp (four complete passes) thì $A = ?$.			
Câu 16.	Cho function sau:			
<pre>int mystery_2(Node *tree = this.root): if(tree == nullptr) return 0;</pre>				
	<pre>else if (tree.getLeft() == nullptr and tree.getRight() == nullptr): return 1; return 1 + mystery_2(tree.getLeft()) + mystery_2(tree.getRight());</pre>			
	Hãy cho biết công dụng của function này:			
Câu 17.	. Chèn các khóa 12, 18, 13, 2, 3, 23, 5 và 15 vào 1 empty hash table có size là 10, sử dụng h(k) = k mod 10 và linear probling. Hãy vẽ hash table kết quả (index từ 0).			
Câu 18.	Cho array $A = [54, 26, 93, 17, 77, 31, 44, 55, 20]$, sử dụng bubble sort sắp xếp A tăng dần , sau 4 bước sắp xế (four complete passes) thì $A = ?$.			
Câu 19.	Cho array $A = [54, 26, 93, 17, 77, 31, 44, 55, 20]$, lần lư empty MaxHeap, hãy vẽ cây Heap cuối cùng được tạo	ượt insert các phần tử của A theo thứ tự index từ 0->7 vào 1 ra.		

Cho hình bên dưới: Head Head Head A New node Head và New node như hình: Thead Cho array A = [21, 1, 26, 45, 29, 28, 2, 9, 16, 49, 39, 27, 43, 34, 46, 40], lần lượt insert các phần tử của A tự index từ 0->15 vào 1 cây empty AVI., hây về cây AVI. cuối cũng được tạo ra. Xóa lần lượt 9, và 29 của cây AVL ở câu trên. Hây vẽ cây AVL sau khi xóa.	-	
Hãy viết các câu lệnh cần thiết để có thể chèn phần tử New node vào vị trí như hình sau, giả sử đã có Head và New node như hình: Head Cho array A = [21, 1, 26, 45, 29, 28, 2, 9, 16, 49, 39, 27, 43, 34, 46, 40], lần lượt insert các phần tử của A tự index từ 0->15 vào 1 cây empty AVL, hầy về cây AVL cuối cùng được tạo ra.	-	
Hãy viết các câu lệnh cần thiết để có thể chèn phần tử New node vào vị trí như hình sau, giả sử đã có Head và New node như hình: Head Cho array A = [21, 1, 26, 45, 29, 28, 2, 9, 16, 49, 39, 27, 43, 34, 46, 40], lần lượt insert các phần tử của A tự index từ 0->15 vào 1 cây empty AVL, hầy về cây AVL cuối cùng được tạo ra.	-	
Hãy viết các câu lệnh cần thiết để có thể chèn phần tử New node vào vị trí như hình sau, giả sử đã có Head và New node như hình: Head Cho array A = [21, 1, 26, 45, 29, 28, 2, 9, 16, 49, 39, 27, 43, 34, 46, 40], lần lượt insert các phần tử của A tự index từ 0->15 vào 1 cây empty AVL, hầy về cây AVL cuối cùng được tạo ra.	-	
Hãy viết các câu lệnh cần thiết để có thể chèn phần tử New node vào vị trí như hình sau, giả sử đã có Head và New node như hình: Head Cho array A = [21, 1, 26, 45, 29, 28, 2, 9, 16, 49, 39, 27, 43, 34, 46, 40], lần lượt insert các phần tử của A tự index từ 0->15 vào 1 cây empty AVL, hầy về cây AVL cuối cùng được tạo ra.	-	
Hãy viết các câu lệnh cần thiết để có thể chèn phần tử New node vào vị trí như hình sau, giả sử đã có Head và New node như hình: Head Cho array A = [21, 1, 26, 45, 29, 28, 2, 9, 16, 49, 39, 27, 43, 34, 46, 40], lần lượt insert các phần tử của A tự index từ 0->15 vào 1 cây empty AVL, hầy về cây AVL cuối cùng được tạo ra.	-	
Hay viết các câu lệnh cần thiết để có thể chèn phần tử New node vào vị trí như hình sau, giả sử đã có Head và New node như hình: Head Cho array A = [21, 1, 26, 45, 29, 28, 2, 9, 16, 49, 39, 27, 43, 34, 46, 40], lần hượt insert các phần tử của A tự index từ 0->15 vào 1 cây empty AVL, hãy vẽ cây AVL cuối cùng được tạo ra.	1	Cho hình bên dưới:
Hay viết các câu lệnh cần thiết để có thể chèn phần tử New node vào vị trí như hình sau, giả sử đã có Head và New node như hình: Head Cho array A = [21, 1, 26, 45, 29, 28, 2, 9, 16, 49, 39, 27, 43, 34, 46, 40], lần hượt insert các phần tử của A tự index từ 0->15 vào 1 cây empty AVL, hãy vẽ cây AVL cuối cùng được tạo ra.		
Head New node Hãy viết các câu lệnh cần thiết để có thể chèn phần tử New node vào vị trí như hình sau, giả sử đã có Head và New node như hình: Head 15 17 40 40 40 41 40 41 40 41 40 41 40 41 40 41 40 41 41		4 1 15 7 40 1
Hãy viết các câu lệnh cần thiết để có thể chèn phần tử New node vào vị trí như hình sau, giả sử đã có Head và New node như hình: 15		Head
Head và New node như hình: Head 15 17 40 Head Cho array A = [21, 1, 26, 45, 29, 28, 2, 9, 16, 49, 39, 27, 43, 34, 46, 40], lần lượt insert các phần tử của A tự index từ 0->15 vào 1 cây empty AVL, hãy vẽ cây AVL cuối cùng được tạo ra.		
Cho array A = [21, 1, 26, 45, 29, 28, 2, 9, 16, 49, 39, 27, 43, 34, 46, 40], lần lượt insert các phần tử của A tự index từ $0->15$ vào 1 cây empty AVL, hấy về cây AVL cuối cùng được tạo ra.		Head và New node như hình:
Cho array A = [21, 1, 26, 45, 29, 28, 2, 9, 16, 49, 39, 27, 43, 34, 46, 40], lần lượt insert các phần tử của A tự index từ $0->15$ vào 1 cây empty AVL, hấy về cây AVL cuối cùng được tạo ra.		4 15 17 40 dota
Cho array A = [21, 1, 26, 45, 29, 28, 2, 9, 16, 49, 39, 27, 43, 34, 46, 40], lần lượt insert các phần tử của A tự index từ 0 ->15 vào 1 cây empty AVL, hãy vẽ cây AVL cuối cùng được tạo ra.		10 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
tự index từ 0->15 vào 1 cây empty AVL, hãy vẽ cây AVL cuối cùng được tạo ra.		Head
tự index từ 0->15 vào 1 cây empty AVL, hãy vẽ cây AVL cuối cùng được tạo ra.	-	
tự index từ 0->15 vào 1 cây empty AVL, hãy vẽ cây AVL cuối cùng được tạo ra.	-	
tự index từ 0->15 vào 1 cây empty AVL, hãy vẽ cây AVL cuối cùng được tạo ra.	-	
tự index từ 0->15 vào 1 cây empty AVL, hãy vẽ cây AVL cuối cùng được tạo ra.	-	
tự index từ 0->15 vào 1 cây empty AVL, hãy vẽ cây AVL cuối cùng được tạo ra.	-	
tự index từ 0->15 vào 1 cây empty AVL, hãy vẽ cây AVL cuối cùng được tạo ra.	-	
	í	Cho array $A = [21, 1, 26, 45, 29, 28, 2, 9, 16, 49, 39, 27, 43, 34, 46, 40]$, lần lượt insert các phần tử của A
Xóa lần lượt 9, và 29 của cây AVL ở câu trên. Hãy vẽ cây AVL sau khi xóa.	1	tự index từ 0->15 vào 1 cây empty AVL, hay ve cây AVL cuối cũng được tạo ra.
Xóa lần lượt 9, và 29 của cây AVL ở câu trên. Hãy vẽ cây AVL sau khi xóa.	-	
Xóa lần lượt 9, và 29 của cây AVL ở câu trên. Hãy vẽ cây AVL sau khi xóa.	-	
Xóa lần lượt 9, và 29 của cây AVL ở câu trên. Hãy vẽ cây AVL sau khi xóa.	-	
Xóa lần lượt 9, và 29 của cây AVL ở câu trên. Hãy vẽ cây AVL sau khi xóa.	-	
Xóa lần lượt 9, và 29 của cây AVL ở câu trên. Hãy vẽ cây AVL sau khi xóa.		
Xóa lần lượt 9, và 29 của cây AVL ở câu trên. Hãy vẽ cây AVL sau khi xóa.		
Xóa lần lượt 9, và 29 của cây AVL ở câu trên. Hãy vẽ cây AVL sau khi xóa.		
		Xóa lần lượt 9, và 29 của cây AVL ở câu trên. Hãy vẽ cây AVL sau khi xóa.
		Xóa lần lượt 9, và 29 của cây AVL ở câu trên. Hãy vẽ cây AVL sau khi xóa.

của A theo thứ

III. LẬP TRÌNH (5 điểm)

Câu 25. (1d) (CODING STYLE)

Sinh viên sẽ có điểm câu này nếu làm ít nhất 1 câu trong các câu tiếp theo và có coding style tốt. Điểm được chấm theo 3 mức: 0, 0.5, hoặc 1.

- Câu 26. Viết function bool moveK2first(Node* head, int k) để di chuyển node ở vị trí thứ k ra đầu danh sách. Giả sử phần tử đầu tiên nằm ở vị trí 1.
- Câu 27. Viết function int countLeave(TreeNode* root) để đếm số lá có trong 1 cây nhi phân tổng quát.
- Câu 28. Cho các khai báo sau:

```
struct StudentNode {
    int id;
    string name;
    StudentNode* next;
};

struct ClassNode {
    int id;
    string name;
    StudentNode* students;
    ClassNode* next;
};
```

Trong đó, cấu trúc Student Node dùng để lưu trữ một node trong danh sách liên kết đơn chứa thông tin sinh viên của một lớp, Class Node dùng để lưu trữ một node trong danh sách liên kết đơn chứa thông tin của các lớp đang quản lý trong một trường học. Thành phần students của Class Node đại diện cho danh sách sinh viên của lớp đó.

Hãy định nghĩa hàm bool addStudent(ClassNode* list, int classId, int studentId, string studentName), trong đó list là con trỏ trỏ đến phần tử đầu tiên trong danh sách lớp, classId, studentId, studentName lần lượt là mã số lớp, mã số sinh viên và tên của sinh viên. Hàm thực thực hiện chức năng sau:

- Thêm sinh viên vào đúng lớp có class Id tương ứng và trả về true. Nếu không tồn tại class Id đó, không thực hiện và trả về false.
- Nếu tồn tai studentId (trong toàn bô các lớp) không thực hiện và trả về false.
- Câu 29. Viết function void printSmallerNearestNode(BSTNode* root, int key) để tìm kiếm và in ra màn hình node duy nhất có giá trị nhỏ (hoặc bằng) hơn và gần nhất với giá trị key trong 1 BST (giá trị key vì thế có thể không tồn tại trong cây).

Họ tên: Trang 4/6 – Mã đề 9