

# ÔN TẬP

Môn: Cấu trúc dữ liệu và Giải thuật

Câu 1: Sắp xếp các hàm sau đây theo thứ tự tăng dần của tốc độ tăng trưởng (khi  $n$  tăng):

$n, \sqrt{n}, n^{1.5}, n \log n, 2/n, 37, 2^n$ .

- ☒ A.  $2/n, 37, \sqrt{n}, n, n \log n, n^{1.5}, 2^n$   
☒ B.  $37, 2/n, \sqrt{n}, n, n \log n, n^{1.5}, 2^n$   
C.  $2/n, 37, \sqrt{n}, n, n^{1.5}, n \log n, 2^n$   
☒ D. Cả A, B, C đều sai
- $n \log n \sim n^{1.5}$   
 $\Rightarrow n \log n = n \cdot n^{1.2}$   
 $\Rightarrow \log n = n^{1/2}$   
 $\Rightarrow \log^2 n = n$   
 $\Rightarrow n \log n$  nhanh hơn

Câu 2. Xét một vector  $a$  đang chứa các phần tử  $\{ 6, 5, 8, 2, 9, 7 \}$  (bắt đầu từ chỉ số 0) và các lệnh sau:

```
for (int i = size; i > 3; i--) // size là kích thước của vector a
    a[i] = a[i - 1];
a[3] = 10;
size++;
```

Sau thao tác các lệnh trên, kết quả dãy  $a$  sẽ là:

- A. 6, 5, 10, 2, 9, 7  
B. 6, 5, 8, 10, 9, 7  
☒ C. 6, 5, 8, 10, 2, 9, 7  
D. 6, 5, 10, 8, 2, 9, 7

Câu 5. Thao tác chèn/xóa ở đầu vector và thao tác chèn/xóa cuối vector có độ phức tạp thời gian lần lượt là:

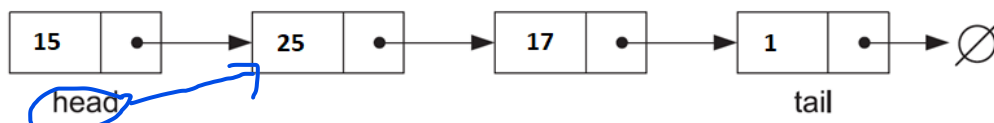
- A.  $O(n)$  và  $O(n)$   
☒ B.  $O(n)$  và  $O(1)$   
C.  $O(1)$  và  $O(n)$   
D.  $O(1)$  và  $O(1)$

Câu 6. Cho danh sách liên kết đơn bao gồm các Node, mỗi Node bao gồm ( $elem, next$ ) trong đó  $elem$  là giá trị,  $next$  là con trỏ trỏ vào nút kế tiếp. Con trỏ vào đầu danh sách là  $head$ . Để chèn một nút mới vào đầu danh sách ta thực hiện như sau

A. Node * v = new Node; v.elem = e; // e là biến cùng kiểu với elem v.next = head;	B. Node * v = head; head = head->next; delete v;
--	--

head = v;	
C.    Node *v = new Node; v.elem = e; v.next = NULL; if (head == NULL) head = v; Node *p = head; while(p->next!=NULL) p= p->next; p->next = v;	<b>D.</b> Cả 3 phương án A, B, C đều sai

Câu 7. Cho danh sách liên kết đơn bao gồm các Node như hình dưới, mỗi Node bao gồm (*elem*, *next*) trong đó *elem* là giá trị, *next* là con trỏ trỏ vào nút kế tiếp. Con trỏ đầu danh sách là *head*.



Với danh sách cụ thể ở trên, hàm sau đây sẽ trả về giá trị nào khi index = 2?

T &operator[] (int index){

```

Node *p = head;
for (int i=1; i<index; i++) p= p->next;
return p->elem;
}

```

A. 15	<b>B. 25</b>	C. 17	D. 1
-------	--------------	-------	------

Câu 9. Cấu trúc dữ liệu nào tương ứng với LIFO

- A. Queue
- B. Danh sách liên kết
- C. Cây
- D. Ngăn xếp**

Câu 10. Cho hai ngăn xếp như sau:

25
21
59

Hình a

15
25
41
59

Hình b

Dãy các thao tác push(x) và pop phải thực hiện để biến đổi hình a thành hình b là:

- A.** pop(), pop(), push(41), push(25), push(15)
- B. push(41), push(25), push(15), pop(), pop()
- C. pop(), push(41), push(25), pop(), push(15)
- D. push(15), pop(), pop(), push(41), push(25)

Câu 12. Cho hai hàng đợi như sau:

20	30		
front	back		

Hình a

28	30	42	17
back		front	

Hình b

Dãy thao tác enqueue(x) và dequeue() để chuyển đổi hàng đợi hình a thành hình b là:

- A.** enqueue(42), enqueue(17), dequeue(), dequeue(), enqueue(28).
- B. dequeue(), dequeue(), enqueue(28), enqueue(17), enqueue(42).
- C. enqueue(28), enqueue(42), dequeue(), dequeue(), enqueue(17).
- D. enqueue(17), enqueue(28), enqueue(42), dequeue(), dequeue().

Câu 14: Hãy cho biết trong các cấu trúc dữ liệu dưới đây, cấu trúc dữ liệu nào có nguyên lý hoạt động là “Vào sau ra trước”

- A.** Ngăn xếp (Stack)
- B. Hàng đợi (Queue)
- C. Danh sách liên kết (Link List)
- D. Cây nhị phân (Binary Tree)

Câu 15: Hãy cho biết trong các cấu trúc dữ liệu dưới đây, cấu trúc dữ liệu nào có nguyên lý hoạt động là “Vào trước ra trước”

- A. Ngăn xếp (Stack)
- B.** Hàng đợi (Queue)
- C. Danh sách liên kết (Link List)
- D. Cây nhị phân (Binary Tree)

Câu 22: Ngăn xếp được ứng dụng trong thuật toán “Tính giá trị của một biểu thức hậu tố”. Hãy cho biết giá trị tại đỉnh của ngăn xếp sau khi thực hiện thuật toán “Tính giá trị của một biểu thức hậu tố” với input là: 3 4 2 + \* 6 -

- A. 10
- B.** 12
- C. -12
- D. 17

Câu 24: Cho Stack có các phép toán:

push(X): Thêm phần tử X vào Stack

pop() : Lấy 1 phần tử ra khỏi Stack

Hãy cho biết phần tử ở đỉnh của Stack có giá trị bằng bao nhiêu sau khi thực hiện lần lượt các phép toán sau: push(5); push(3); pop(); push(4); push(6); pop()

A. 3

C. 5

**B. 4**

D. 6

Câu 26: Cho Queue có các phép toán:

EnQueue(X): Thêm phần tử X vào Queue

DeQueue() : Lấy 1 phần tử ra khỏi Queue

Hãy cho biết phần tử ở đầu của Queue có giá trị bằng bao nhiêu sau khi thực hiện lần lượt các phép toán sau: EnQueue(5); EnQueue(3); DeQueue(); EnQueue(4); EnQueue(6);

**A. 3**

C. 5

B. 4

D. 6

Câu 28: Cho Queue có các phép toán:

EnQueue(X): Thêm phần tử X vào Queue

DeQueue() : Lấy 1 phần tử ra khỏi Queue

Hãy cho biết phần tử ở đầu của Queue có giá trị là ký tự nào, sau khi thực hiện thuật toán dưới đây với input là: "This\*\*is\*\*\*Queue\*\*"

Thuật toán

Input: Xâu S

Đọc lần lượt từng ký tự từ trái qua phải của xâu S; Nếu ký tự đọc được là '\*' thì lấy 1 phần tử ra khỏi Queue. Ngược lại thì thêm phần tử đọc được vào Queue.

A. T

C. u

B. h

**D. Q**

Câu 29: Cho Stack có các phép toán:

push(X): Thêm phần tử X vào Stack

pop() : Lấy 1 phần tử ra khỏi Stack

Hãy cho biết phần tử ở đỉnh của Stack có giá trị là ký tự nào, sau khi thực hiện thuật toán dưới đây với input là: "This\*\*is\*\*\*Stack\*\*"

Thuật toán

Input: Xâu S

Đọc lần lượt từng ký tự từ trái qua phải của xâu S; Nếu ký tự đọc được là '\*' thì lấy 1 phần tử ra khỏi Stack. Ngược lại thì thêm phần tử đọc được vào Stack.

A. T

**C. c**

B. S

D. k

Câu 30: Cây nhị phân là cây mà mỗi nút trên cây có .....

A. Hai cây con

B. Tối thiểu hai cây con

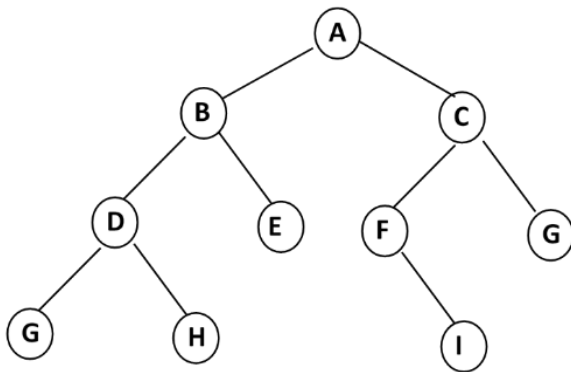
☒ C. Tối đa hai cây con

D. Có một hoặc hai cây con

Câu 32: Thứ tự nào sau đây cho phép duyệt đệ quy cây nhị phân theo thứ tự trước

- A. Duyệt cây con trái theo thứ tự trước -> thăm gốc -> duyệt cây con phải theo thứ tự trước
- B. Duyệt cây con trái theo thứ tự trước -> duyệt cây con phải theo thứ tự trước -> thăm gốc
- ☒ C. Thăm gốc -> duyệt cây con trái theo thứ tự trước -> duyệt cây con phải theo thứ tự trước
- D. Thăm gốc -> duyệt cây con phải theo thứ tự trước -> duyệt cây con trái theo thứ tự trước

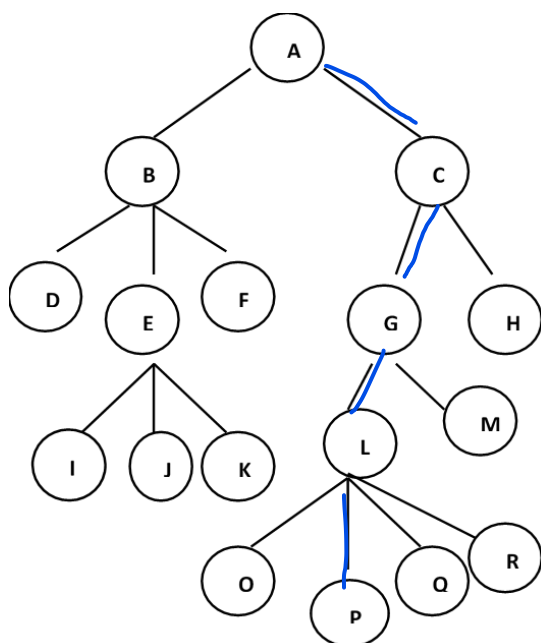
Câu 33: Cho cây nhị phân như hình vẽ sau:



Hãy cho biết dãy các nút theo thứ tự duyệt giữa

- A. A B D G H E C F I G
- B. G H D E B I F G C A
- ☒ C. G D H B E A F I C G
- D. G D H B E F I C G

Câu 34: Cho biết chiều cao của cây ở hình vẽ sau:



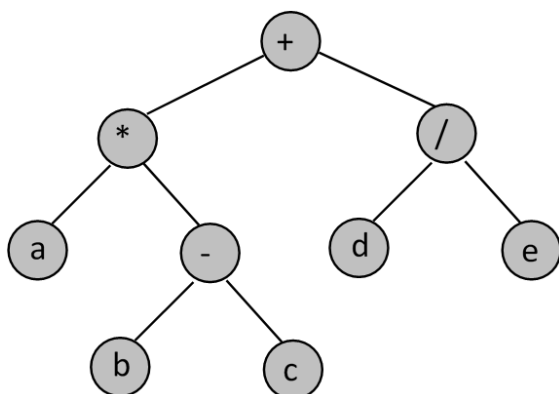
A. 2

B. 3

**C. 4**

D. 5

Câu 35: Cho cây biểu thức sau:



LNR

Biểu thức nào sau đây tương ứng với cây

**A.  $a * (b - c) + d / e$**

B.  $a * b - c + d / e$

C.  $a + (b - c) * d / e$

D.  $b - c * a + d / e$

Câu 36: Trong phép duyệt một cây nhị phân có 24 nút theo thứ tự sau, nút gốc có thứ tự duyệt thứ mấy ?

A. Thứ 1

B. Thứ 2

C. Thứ 23

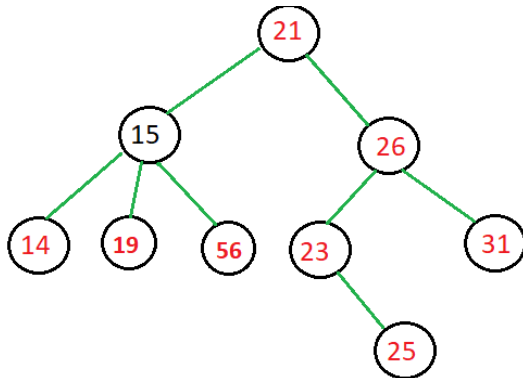
**D. Thứ 24**

Câu 37: Nút có khóa nhỏ nhất trong cây nhị phân tìm kiếm khác rỗng là:

- A. Nút gốc  
B. Tất cả các nút

- C. Nút con bên phải nhất  
**D. Nút con bên trái nhất**

Câu 39.



Cho cây như hình vẽ. Các giá trị nút được duyệt theo thứ tự trước là ...

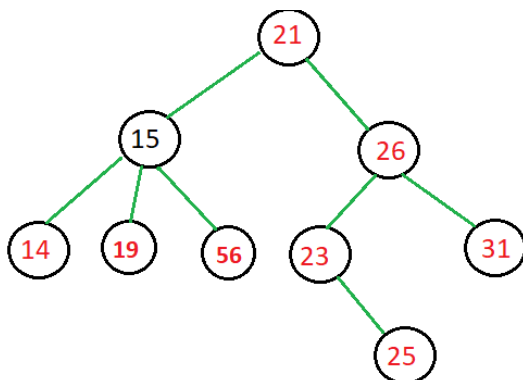
**A.** 21, 15, 14, 19, 56, 26, 23, 25, 31

~~B.~~ 14, 19, 56, 15, 25, 23, 31, 26, 21

~~C.~~ 14, 15, 19, 56, 21, 23, 25, 26, 31

D. Cả 3 phương án A, B, C đều sai

Câu 40.



Cho cây như hình vẽ, các giá trị nút được duyệt theo thứ tự sau là ...

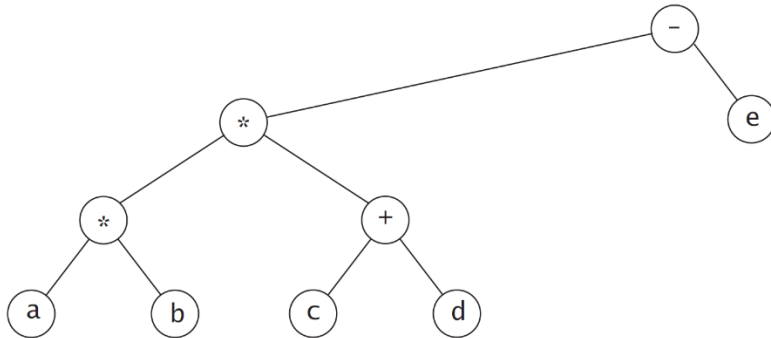
A. 21, 15, 14, 19, 56, 26, 23, 25, 31

B. 21, 14, 15, 19, 56, 25, 26, 23, 31

C. 14, 15, 19, 56, 21, 23, 25, 26, 31

**D.** Cả 3 phương án A, B, C đều sai

Câu 41. Cho cây biểu thức như hình sau:



Biểu thức hậu tố của cây biểu thức là: **L R N**

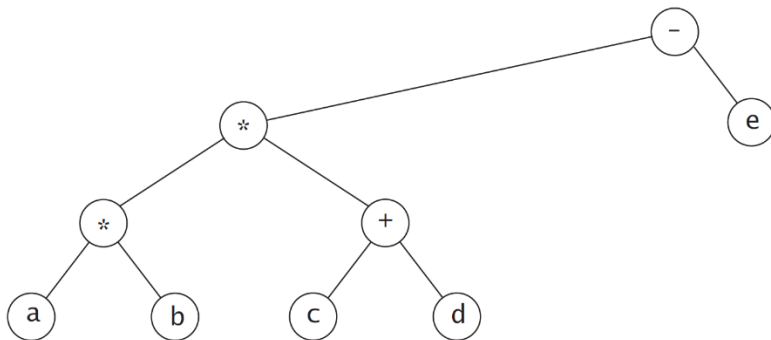
**A.**  $a b * c d + * e -$

**C.**  $- * * a b + c d e$

**B.**  $(a * b) * (c + d) - e$

**D.** cả 3 phương án A, B, C đều sai

Câu 42. Cho cây biểu thức như hình sau:



Biểu thức tiền tố của cây biểu thức là: **N L R**

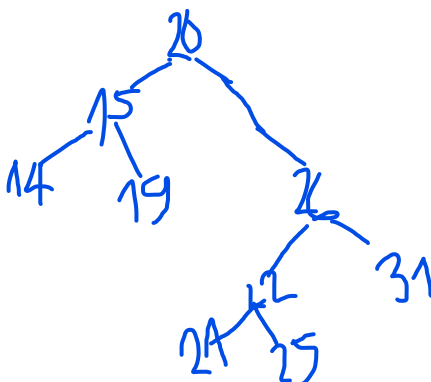
**A.**  $a b * a d + * e -$

**C.**  $- * * a b + c d e$

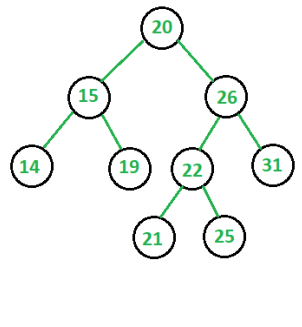
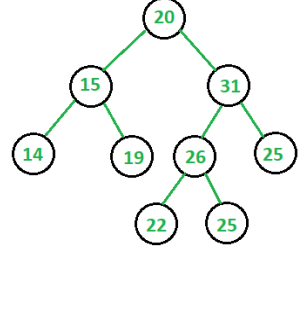
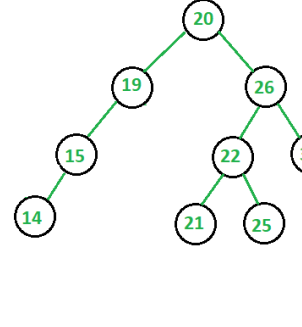
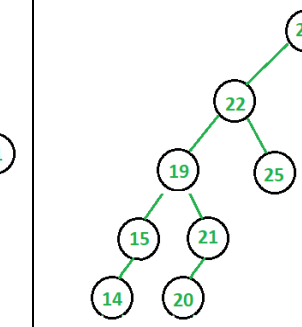
**B.**  $(a * b) * (c + d) - e$

**D.** cả 3 phương án A, B, C đều sai

Câu 43. Khi chèn lần lượt các giá trị { 20, 15, 19, 26, 31, 22, 14, 21, 25 } vào cây nhị phân tìm kiếm ban đầu rỗng. Ta thu được hình ảnh cây nào ?



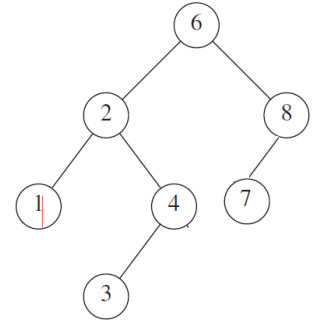


			
<b>A</b>	B	C	D

Câu 44. Cho cây nhị phân tìm kiếm như hình dưới, bao gồm các Node, mỗi Node bao gồm (*elem*, *left*, *right*) trong đó *elem* là giá trị nguyên, *left* là con trỏ trái, *right* là con trỏ phải. Con trỏ gốc là *root*.

Đoạn mã dưới đây về cây sẽ trả về giá trị nào:

```
int * find(Node * root) {
    if(root!= NULL)
        while (root->left!=NULL)
            root = root->left;
    return root->elem;
}
```

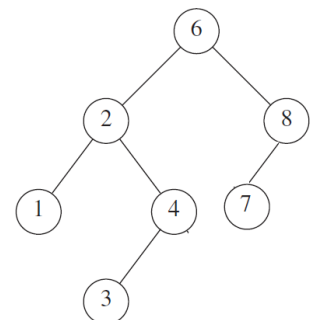


<b>A. 1</b>	B. -1	C. 3	D. 8
-------------	-------	------	------

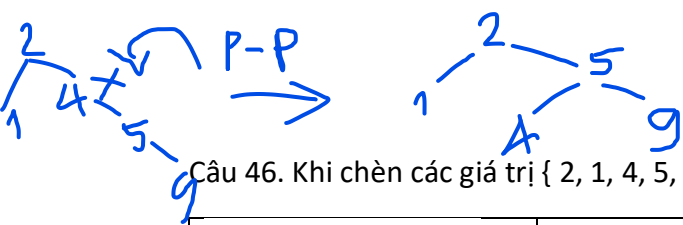
Câu 45. Cho cây nhị phân tìm kiếm như hình dưới, bao gồm các Node, mỗi Node bao gồm (*elem*, *left*, *right*) trong đó *elem* là giá trị nguyên, *left* là con trỏ trái, *right* là con trỏ phải. Con trỏ gốc là *root*.

Đoạn mã dưới đây về cây sẽ trả về giá trị nào:

```
int * find(Node * root) {
    if (root != NULL)
        while (root->right != NULL)
            root = root->right;
    return root->elem;
}
```



A. 1	B. -1	C. 3	<b>D. 8</b>
------	-------	------	-------------



Câu 46. Khi chèn các giá trị { 2, 1, 4, 5, 9 }. Ta thu được cây AVL nào ?

			Cả ba phương án A, B, C đều đúng.
<b>A</b>	B	C	D

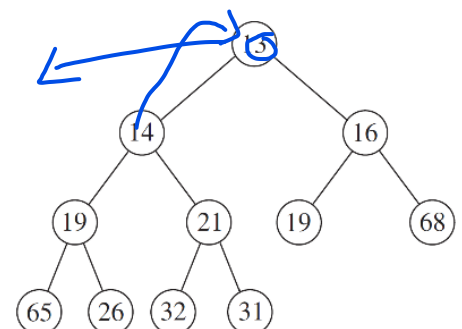
Câu 47. Xét bảng băm đang rỗng và có hàm băm là  $\text{hash}(x) = x \% 10$ . Chèn vào bảng các giá trị { 71, 23, 73, 99, 44, 79, 89 } theo phương pháp thăm dò tuyến tính, ta thu được bảng băm nào?

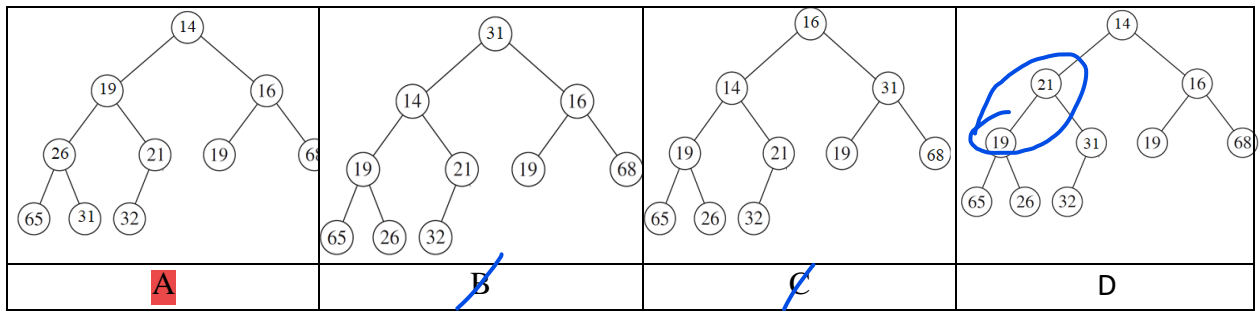
	0	1	2	3	4	5	6	7	8	9
<b>A.</b>	79	71	89	23	73	44				99
B	71	23	73	99	44	79				89
C	79	99	71	73	44	23				89
D	79	71	89	73	23	44				99

Câu 50. Khi chèn các phần tử {10, 12, 1, 14, 6, 5, 8, 15} vào đồng nhị phân (cực tiểu) đang rỗng, ta thu được đồng nhị phân như sau:

<b>A</b>	B	C	D

Câu 51. Khi xóa gốc của đồng nhị phân (hình bên) ta thu được đồng nhị phân mới nào ?





Câu 52. Thuật toán sau đây sắp xếp dãy a gồm n phần tử thành dãy mới có thứ tự tăng dần.

```

for (int i = 0; i < n - 1; i++) {
    int vt = i;
    for (int j = i + 1; j < n; j++)
        if (a[vt] > a[j]) vt = j;
    if (vt != i) {
        tg = a[vt];
        a[vt] = a[i];
        a[i] = tg;
    }
}

```

Thuật toán trên được gọi là thuật toán ...

- A.** Sắp xếp chọn    **selection sort**  
 B. Sắp xếp nổi bọt

- C. Sắp xếp chèn  
 D. Sắp xếp trộn

Câu 54. Cho một dãy a: {15 25 17 1 0} và thuật toán sắp xếp chọn

```

for (int i = 0; i < n - 1; i++) {
    int vt = i;
    for (int j = i + 1; j < n; j++)
        if (a[vt] > a[j]) vt = j;
    if (vt != i) {
        tg = a[vt];
        a[vt] = a[i];
        a[i] = tg;
    }
}

```

Với dãy a cụ thể ở trên, cần thực hiện mấy lần đổi chỗ a[vt] với a[i] để có được dãy a theo thứ tự tăng dần

- A. 2 lần  
 B. 3 lần

- C.** 4 lần  
 D. 5 lần

Câu 55. Giả sử cần sắp xếp mảng A {11, 16, 12, 75, 51, 54, 5, 73, 36, 52, 98 } theo phương pháp sắp xếp chèn trực tiếp. **Số lần chèn** các phần tử vào dãy con L(đã có thứ tự tăng ở đầu dãy) để xếp A theo thứ tự tăng dần là: **so lan chen=n-1**

A. 9 lần

C. 8 lần

**B.** 10 lần

D. 7 lần

Câu 57. Phương pháp nào sau đây được sử dụng để sắp xếp trộn? **merge sort**

**A.** hợp nhất

B. phân vùng

C. lựa chọn

D. trao đổi

**quick sort**

**selection sort**

**bubble sort**

Câu 58. Chọn mã chính xác để sắp xếp trộn ?

a)

```
void merge_sort(int arr[], int left, int right)
{
    if (left > right)
    {
        int mid = (right-left)/2;
        merge_sort(arr, left, mid);
        merge_sort(arr, mid+1, right);
        merge(arr, left, mid, right); //function to merge sorted arrays
    }
}
```

**b)**

```
void merge_sort(int arr[], int left, int right)
{
    if (left < right)
    {
        int mid = left+(right-left)/2;
        merge_sort(arr, left, mid);
        merge_sort(arr, mid+1, right);
        merge(arr, left, mid, right); //function to merge sorted arrays
    }
}
```

$$= \frac{2L+R-L}{2} = \frac{L+R}{2}$$

c)

```
void merge_sort(int arr[], int left, int right)
{
    if (left < right)
    {
        int mid = left+(right-left)/2;
        merge(arr, left, mid, right); //function to merge sorted arrays
        merge_sort(arr, left, mid);
        merge_sort(arr, mid+1, right);
    }
}
```

d)

```
void merge_sort(int arr[], int left, int right)
{
    if (left < right)
    {
        int mid = (right-left)/2;
        merge(arr, left, mid, right); //function to merge sorted arrays
        merge_sort(arr, left, mid);
        merge_sort(arr, mid+1, right);
    }
}
```

Câu 60. Cấu trúc dữ liệu nào sau đây có thời gian tìm kiếm là  $O(1)$  -

- A. Cây
- B. Đồng
- ☒ C. Bảng băm
- D. Danh sách Liên kết

Câu 62. Cần bao nhiêu lần hoán đổi để sắp xếp tăng dãy {2, 5, 3, 7, 1, 4} bằng cách sử dụng sắp xếp nổi bọt?

- A. 5
- B. 6
- ☒ C. 7
- D. 8

Câu 63. Thời gian cần thiết để hợp nhất (trộn) hai danh sách đã sắp xếp có kích thước  $m$  và  $n$ , là

- A.  $O(m / n)$
- ☒ B.  $O(m + n)$
- C.  $O(m \log n)$
- D.  $O(n \log m)$

Câu 64. Một phần tử chốt để phân chia dãy chưa sắp xếp được sử dụng trong ...

- A. Sắp xếp trộn
- ☒ B. Sắp xếp nhanh
- C. Sắp xếp chèn
- D. Sắp xếp lựa chọn

Câu 66. Thuật toán sắp xếp nào sau đây có độ phức tạp trong trường hợp xấu nhất thấp nhất?

- ☒ A. Merge Sort  $n \log n$
- B. Bubble Sort  $n^2$
- C. Quick Sort  $n^2$
- D. Selection Sort  $n^2$

Câu 67. Giả sử chúng ta đang sắp xếp một mảng 8 số nguyên bằng cách sử dụng thuật toán heapsort và chúng ta vừa hoàn thành một số thao tác lấy phần tử ở gốc (max hoặc min). Mảng bây giờ như sau: 16 14 15 10 12 27 28. Hỏi có bao nhiêu phép toán lấy ra phần tử tại gốc của đồng (heap) đã được thực hiện ?

- A. 1
- B. 2

C. 3 hoặc 4

D. 5 hoặc 6

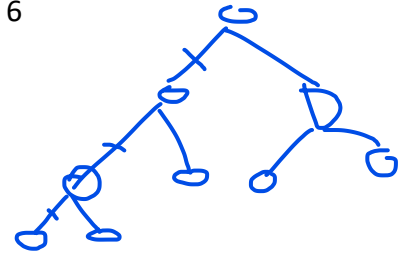
Câu 68. Chiều cao tối đa của cây AVL có 9 nút là bao nhiêu?

A. 2

**B. 3**

C. 4

D. 5



Câu 69: Cần tối đa bao nhiêu phép so sánh để tìm kiếm trên một vectơ đã sắp xếp gồm 1023 phần tử bằng cách sử dụng thuật toán tìm kiếm nhị phân?  $\log_2 n \rightarrow \log_2 n$   $n$

**A. 10**

B. 15

C. 20

D. 30