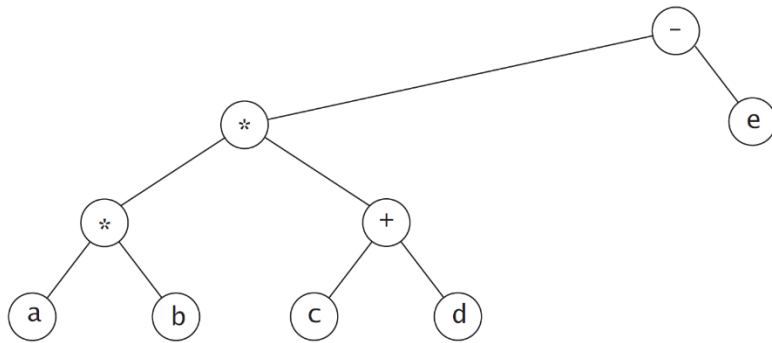


Câu 41. Cho cây biểu thức như hình sau:



Biểu thức hậu tố của cây biểu thức là:

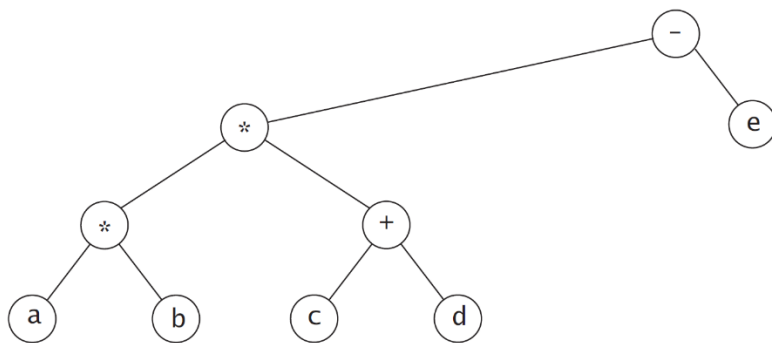
A. $a b * c d + * e -$

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:

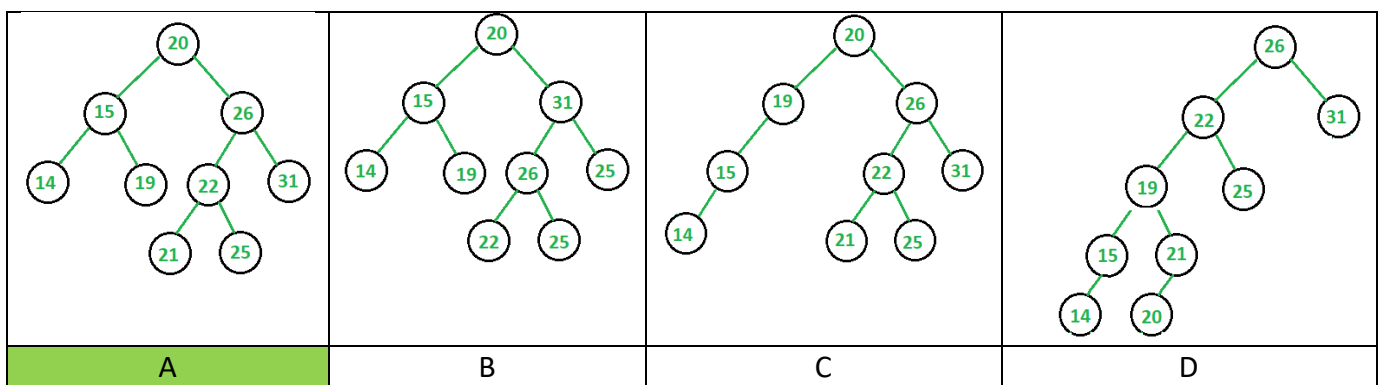
A. $a b * a d + * e -$

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

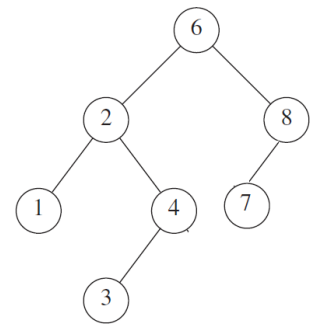
B. $(a * b) * (c + d) - e$

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 như sau:



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*.



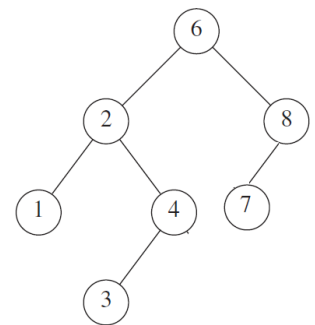
Đoạn mã dưới đây về cây sẽ trả về giá trị nào:

```

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

A. 1	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	D. 8
------	-------	------	------

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

<pre> graph TD 2((2)) --> 1((1)) 2 --> 5((5)) 5 --> 4((4)) 5 --> 9((9)) </pre>	<pre> graph TD 5((5)) --> 2((2)) 5 --> 9((9)) 2 --> 1((1)) 2 --> 4((4)) </pre>	<pre> graph TD 4((4)) --> 2((2)) 4 --> 9((9)) 2 --> 1((1)) 9 --> 5((5)) </pre>	Cả ba phương án A, B, C đều đúng.
A	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 như sau:

	0	1	2	3	4	5	6	7	8	9
A.	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

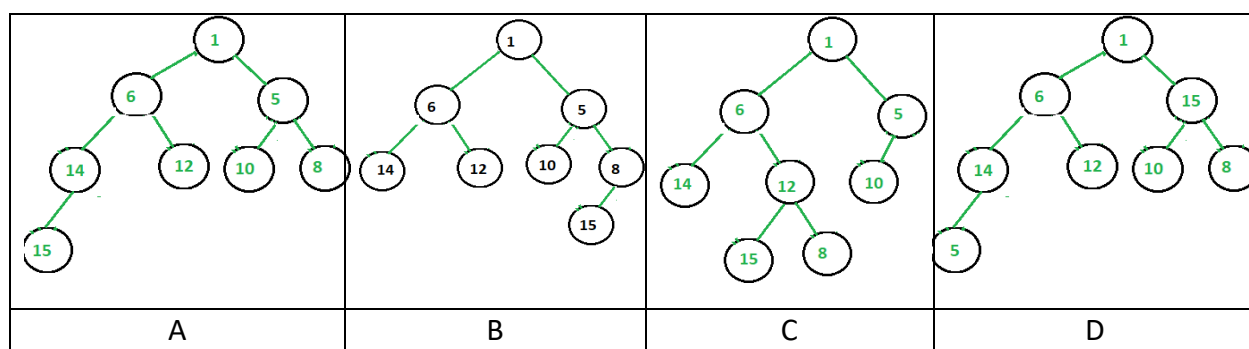
Câu 48. Xét bảng băm đang rỗng và có hàm băm là $\text{hash}(x) = x \% 10$. Chèn vào bảng các giá trị { 71, 23, 73, 99, 44, 79, 89 } theo phương pháp thăm dò bậc 2, ta thu được bảng như sau:

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

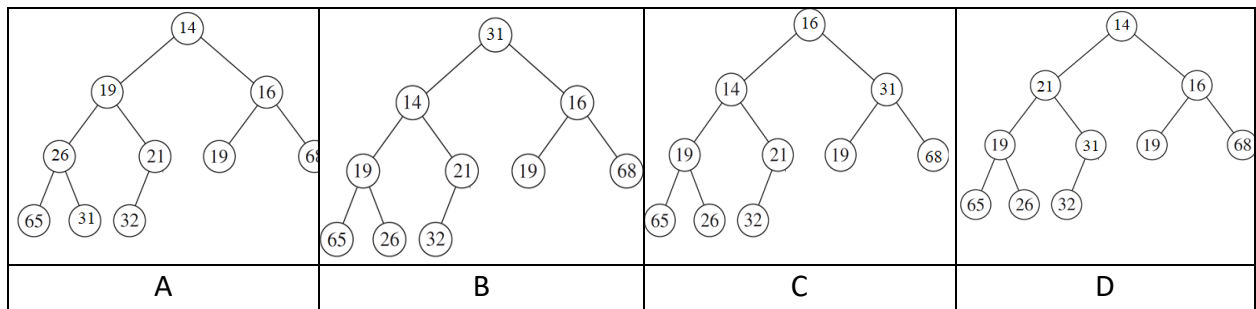
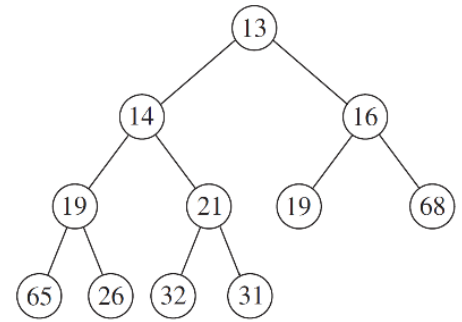
Câu 49. Xét bảng băm đang rỗng và có hàm băm là $\text{hash}(x) = x \% 10$. Chèn vào bảng các giá trị { 71, 23, 73, 98, 43, 79, 89 } theo phương pháp thăm dò bậc 2, ta thu được bảng băm nào ?

	0	1	2	3	4	5	6	7	8	9
A	79	71		23	73			43	98	89
B	89	71		23	73		43		98	79
C	89	71		23	73			43	98	79
D	89	71		23	73	43			98	79

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:



Câu 51. Khi xóa gốc của đồng nhị phân (hình bên) ta thu được được đồng nhị phân mới như sau:



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

A. Sắp xếp chọn

B. Sắp xếp nổi bọt

C. Sắp xếp chèn

D. Sắp xếp trộn

Câu 53. 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++) {
    for (int j = 0; j < n-1-i; j++) {
        if (a[j] > a[j+1]) {
            T tg = a[j];
            a[j] = a[j+1];
            a[j+1] = tg;
        }
    }
}

```

}

Thuật toán trên được gọi là

A. Sắp xếp chọn

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:

A. 9 lần

B. 10 lần

C. 8 lần

D. 7 lần

Câu 56. Điều nào sau đây là một lợi thế của cây AVL so với đồng nhị phân?

A) chèn mất ít thời gian hơn

B) xóa mất ít thời gian hơn

C) tìm kiếm mất ít thời gian hơn

D) việc xây dựng cây mất ít thời gian hơn so với đồng nhị phân

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

A. hợp nhất

B. phân vùng

C. lựa chọn

D. trao đổi

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
    }
}
```

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 59. Điều gì không đúng về sắp xếp chèn?

- A. hiệu suất kém nhất khi mảng ban đầu được sắp xếp theo thứ tự ngược lại.
- B. Trường hợp xấu nhất và trường hợp trung bình là $O(n^2)$
- C. Có thể so sánh với cách một người chơi bài sắp xếp quân bài của mình từ một bộ bài.
- D. Không có điều nào ở trên.

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

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

Câu 61. Bạn phải sắp xếp một danh sách L bao gồm: một danh sách đã sắp xếp và một vài phần tử ngẫu nhiên nào đó. Phương pháp sắp xếp nào sau đây sẽ đặc biệt phù hợp cho một nhiệm vụ như vậy?

- A. Sắp xếp nổi bọt
- B. Sắp xếp lựa chọn
- C. Sắp xếp nhanh
- D. Sắp xếp chèn

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

- A. 4
- B. 5
- C. 6
- D. 7

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

- A. Merge Sort
- B. Bubble Sort
- C. Quick Sort
- D. Selection Sort

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 đống (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 vector đã sắp xếp gồm 1023 phần tử bằng cách sử dụng thuật toán tìm kiếm nhị phân?

- A. 10
- B. 15
- C. 20
- D. 30

Câu 70: Thuật toán sắp xếp nào là tốt nhất nếu danh sách đã có thứ tự?

- A. Sắp xếp nhanh (Quick Sort)
- B. Sắp kiểu trộn (Merge Sort)
- C. Sắp xếp chèn (Insertion Sort)
- D. Không có