1-1

If the inorder and the postorder traversal sequences of a binary tree have exactly the same order, then none of the nodes in the tree has a right subtree. (2 分)

T F

作者: 干红华

单位: 浙江大学

1-2

If N numbers are stored in a doubly linked list in increasing order, then the average time complexity for binary search is $O(logN)$. (2 分)

T F

作者: 陈翔

单位: 浙江大学

1-3

To sort N records by quick sort, the worst-case time complexity is $O(NlogN)$. (2 分)

T F

作者: DS 课程组

单位: 浙江大学

1-4

Let M be the minimum spanning tree of a weighted graph G. Then the path in M between V1 and V2 must be the shortest path between them in G. (2 分)

T F

作者: 陈越

单位: 浙江大学

1-5

For a graph, if each vertex has an even degree, we can find an Euler circuit that visits every vertex exactly once. (2 分)

T F

作者: 何钦铭

单位: 浙江大学

1-6

Quadratic probing is equivalent to double hashing with a secondary hash function of $Hash_2(k)=k$. (2 分)

T F

作者: 朱建科

单位: 浙江大学

1-7

$logN^{20}$ is $O(N)$. (2 分)

T F

作者: 陈翔

单位: 浙江大学

1-8

Given a binary search tree with 20 integer keys which include 4, 5, and 6, if 4 and 6 are on the same level, then 5 must be their parent. (2 分)

T F

作者: 干红华

单位: 浙江大学

1-9

Mergesort is stable. (2 分)

T F

作者: fengyan

单位: 浙江大学

1-10

If keys are pushed onto a stack in the order `abcde`, then it's impossible to obtain the output sequence `cdabe`. (2 分)

T F

作者: 陈翔

单位: 浙江大学

2-1

For the quicksort implementation with both the left and the right pointers stop when an element with the same key as the pivot is found during the partitioning, what is the running time when all keys are equal? (3 分)

A. $O(NlogN)$

B. $O(N)$

C. $O(N2)$

D. $O(logN)$

作者: DS 课程组

单位: 浙江大学

2-2

Given a tree of degree 6. Suppose that the numbers of nodes of degrees 1, 2, 3, 4, 5, 6 are 3, 5, 3, 4, 2, 1, respectively. Then the number of leaf nodes must be: (3 分)

A. 33

B. 31

C. 37

D. 35

作者: 干红华

单位: 浙江大学

2-3

A graph with 100 vertices and 12 edges must have at most __ connected component(s). (3 分)

A. 88

B. 95

C. 94

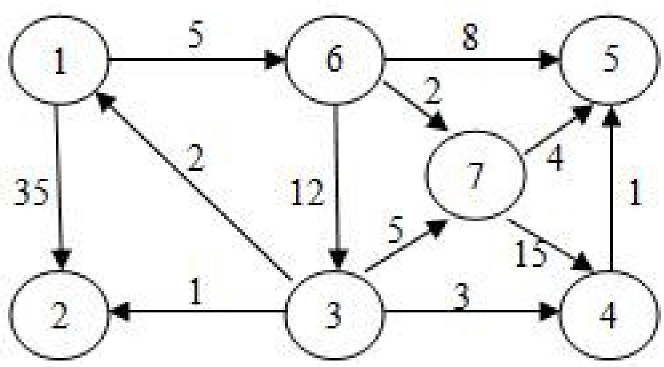
D. 87

作者: 陈越

单位: 浙江大学

2-4

Use Dijkstra algorithm to find the shortest paths from 1 to every other vertices. In which order that the destinations must be obtained? (3 分)



A. 2, 4, 3, 6, 5, 7

B. 2, 3, 4, 5, 6, 7

C. 6, 2, 5, 7, 3, 4

D. 6, 7, 5, 3, 2, 4

作者: 陈越

单位: 浙江大学

2-5

For an in-order threaded binary tree, if the pre-order and in-order traversal sequences are `D A B C F E`and `B A C D E F` respectively, which pair of nodes' right links are both threads? (3 分)

A. B and E

B. B and A

C. A and F

D. D and A

作者: 干红华

单位: 浙江大学

2-6

The inorder and the postorder traversal sequences of a binary tree are `a b c d e f g` and `a c b f g e d`, respectively. Then the preorder traversal sequences is: (3 分)

A. `d a c b f e g`

B. `d c a b e g f`

C. `d b a c e g f`

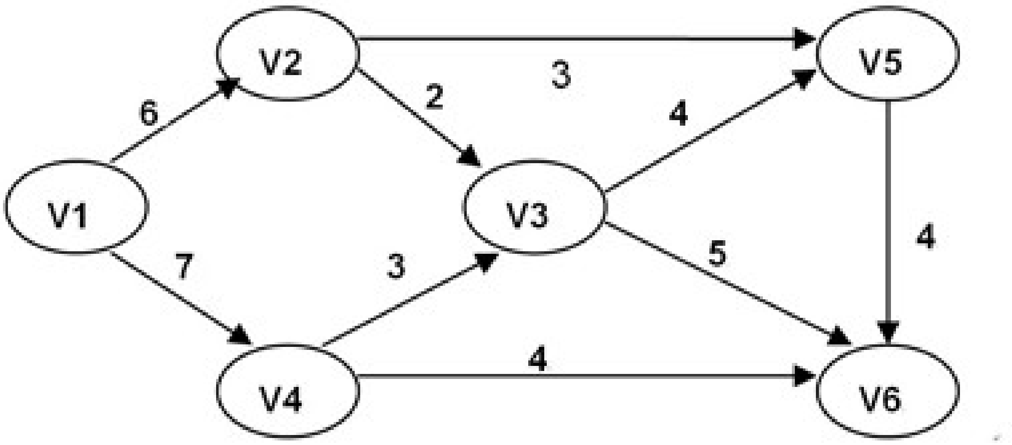
D. `d b a c f e g`

作者: 干红华

单位: 浙江大学

2-7

The maximum flow from v1 to v6 is __ (3 分)



A. 0

B. 13

C. 12

D. 11

作者: 何钦铭

单位: 浙江大学

2-8

When inserting a new key `K` into a binary search tree `T` with 512 nodes, the worst-case number of comparisons between `K` and the keys already in `T` is in the range of: (3 分)

A. [9, 512]

B. [10, 512]

C. [9, 511]

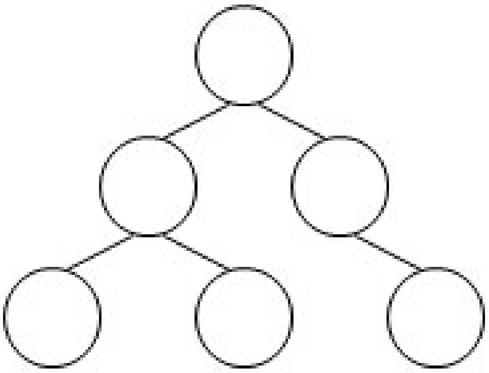
D. [10, 511]

作者: 干红华

单位: 浙江大学

2-9

Given the structure of a binary search tree (as shown in the figure), which one of the following insertion sequences is impossible? (3 分)



A. `83 91 98 67 75 20`

B. `83 67 75 91 20 98`

C. `83 91 75 67 20 98`

D. `83 67 91 98 20 75`

作者: 干红华

单位: 浙江大学

2-10

It is known that a 3-heap is a heap whose nodes have 3 children. Suppose that the level-order traversal sequence of a max-3-heap is {88, 76, 65, 82, 68, 46, 52, 44, 62, 33, 75, 28, 55, 60}. Use the linear algorithm to adjust this max-3-heap into a min-3-heap, and then run DeleteMin. As a result, there are __ nodes whose positions are not moved in the process. (3 分)

A. 5

B. 4

C. 3

D. 2

作者: 徐镜春

单位: 浙江大学

2-11

Given input { 46, 79, 56, 38, 40, 84 }. Which one of the following is the initial heap built by heap sort? (3 分)

A. 84, 79, 56, 38, 40, 46

B. 84, 79, 56, 46, 40, 38

C. 84, 56, 79, 40, 46, 38

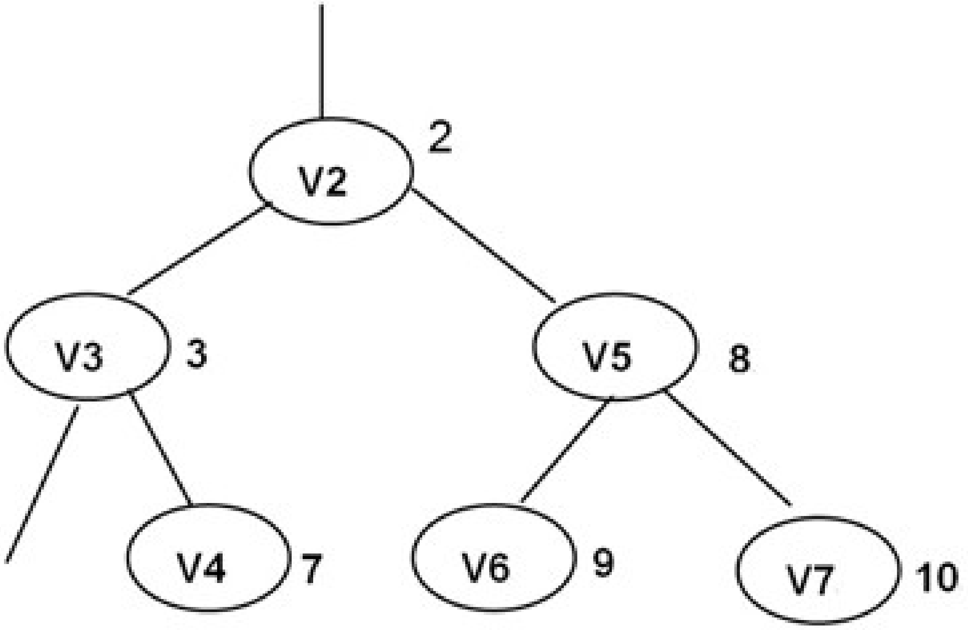
D. 79, 46, 56, 38, 40, 80

作者: DS 课程组

单位: 浙江大学

2-12

The following is the part of depth-first search tree to find the articulation points, and the Num(v) value has been marked beside each vertex v. The back edges are not shown. Which of the following situation is impossible? (3 分)



A. low(v7) is 2

B. low(v6) is 3

C. low(v4) is 2

D. low(v6) is greater than low(v7)

作者: 何钦铭

单位: 浙江大学

2-13

Following is the C-like pseudo code of a function that takes a Queue as an argument.

```
void foo(Queue Q)
{
    Queue Q1 = CreateQueue();  // create an empty queue

    while (!IsEmpty(Q))
    {
        // dequeue an item from Q and enqueue it into Q1
        Enqueue(Q1, Dequeue(Q));
    }

    while (!IsEmpty(Q1))
    {
        // dequeue an item from Q1 and enqueue it into Q
        Enqueue(Q, Dequeue(Q1));
    }

    DisposeQueue(Q1);
}
```

What does the above function do? (3 分)

A. Makes Q empty

B. Reverses Q

C. Keeps Q unchanged

D. Removes the last item from Q

作者: 陈翔

单位: 浙江大学

2-14

Which one of the following is a possible postorder traversal sequence of a binary search tree? (3 分)

A. `2 1 4 5 3 10 7 9 8 6`

B. `2 1 4 5 3 7 10 9 8 6`

C. `2 4 1 5 3 7 10 9 8 6`

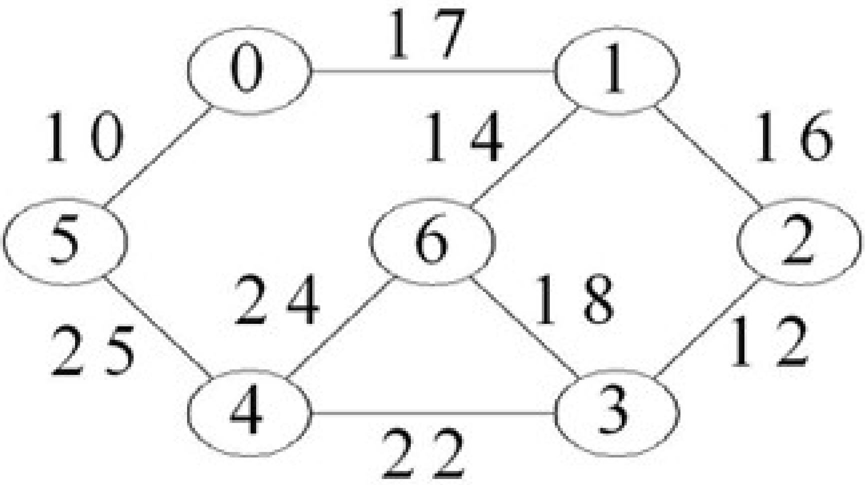
D. `2 4 1 5 3 7 9 10 8 6`

作者: 干红华

单位: 浙江大学

2-15

To find the minimum spanning tree with Prim's algorithm for the following graph, a sequence of vertexes 6, 1, 2, 3 was found during the algorithm's early steps. Which one vertex will be added in the next step? (3 分)



A. 5

B. 4

C. the vertex serial is incorrect

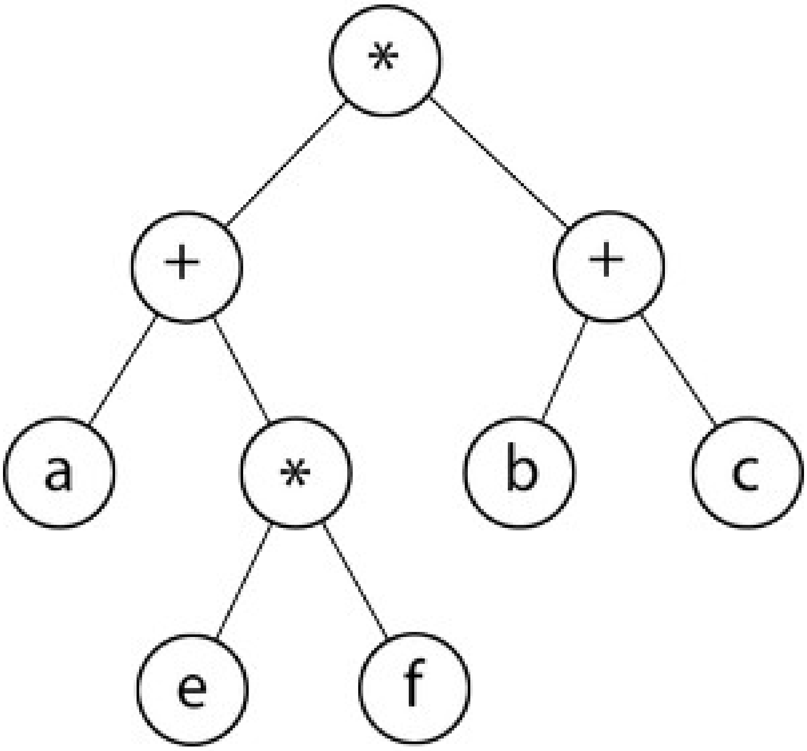
D. 0

作者: 何钦铭

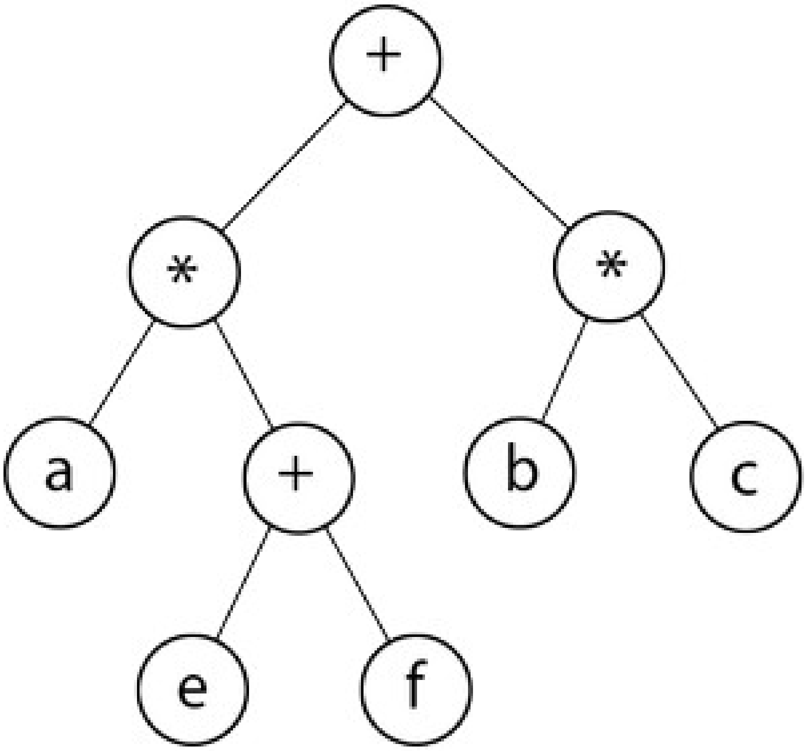
单位: 浙江大学

2-16

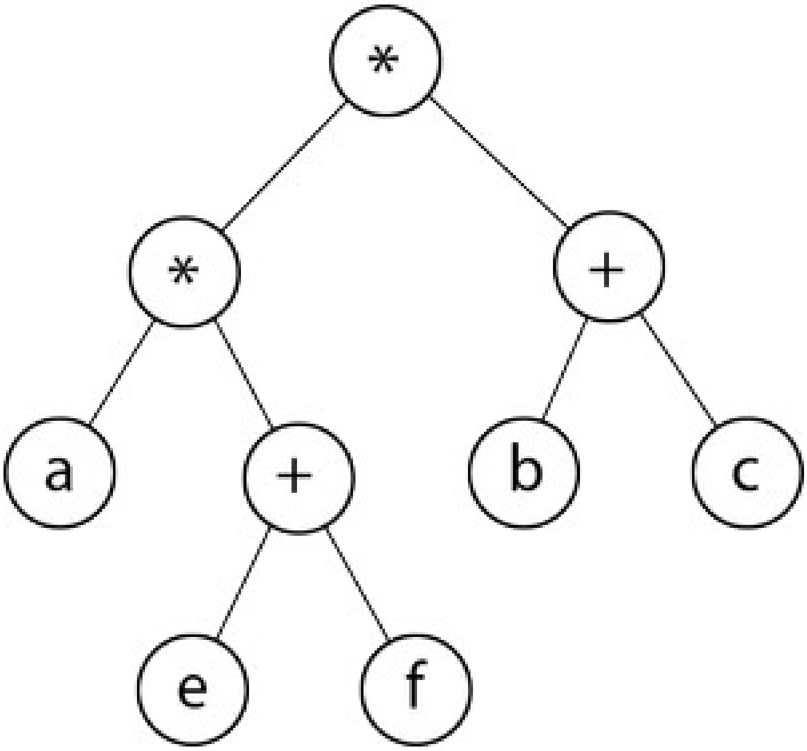
Which one of the following is the expression tree corresponding to the postfix expression `aef*+bc*+`? (3 分)



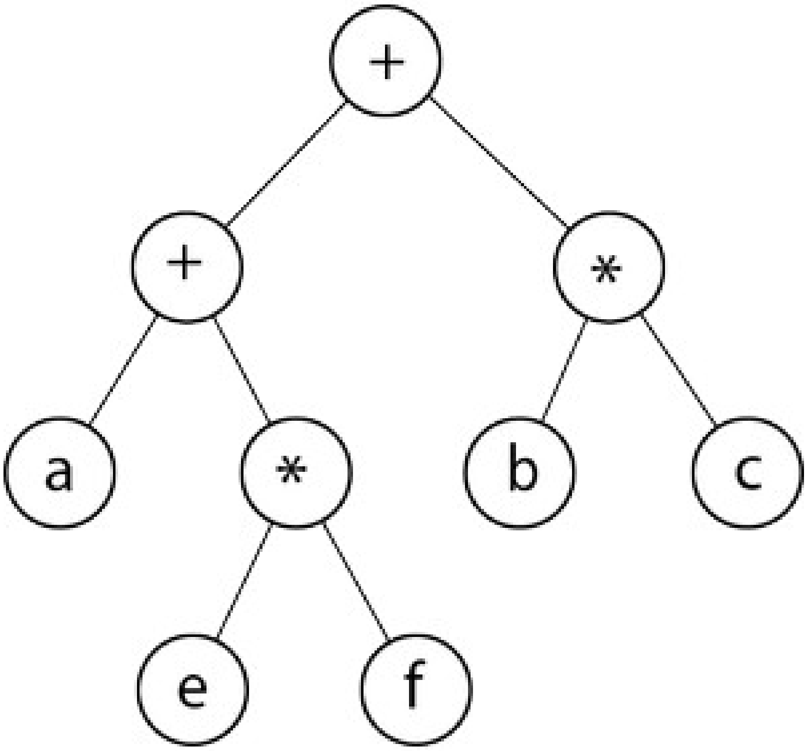
A.



B.



C.



D.

作者: 干红华

单位: 浙江大学

2-17

Suppose that the size of a hash table is 11, and the hash function is H(key)=key%11. The following 4 elements have been inserted into the table as Addr(14)=3, Addr(38)=5, Addr(61)=6, Addr(86)=9. When open addressing with quadratic probing is used to solve collisions, the address of the element with key=49 will be . (3 分)

A. 8

B. 7

C. 10

D. 5

作者: 朱建科

单位: 浙江大学

2-18

In order to convert the infix expression `4 * 3 + (6 * 3 - 12)` to postfix expression using a stack S, then the minimum size of S must be: (3 分)

A. 5

B. 4

C. 3

D. 2

作者: 陈翔

单位: 浙江大学

2-19

Let T be a tree of N nodes created by union-by-size without path compression, then the minimum depth of T may be (3 分)

A. $N/2$

B. $N-1$

C. $logN$

D. 1

作者: 朱建科

单位: 浙江大学

2-20

Insert {18, 23, 4, 26, 31, 33, 17, 39} one by one into an initially empty hash table of size 13 with the hash function $H(Key)=Key\%13$, and linear probing is used to resolve collisions. What is the loading density when the first collision occurs? (3 分)

A. 0.31

B. 0.62

C. 0.63

D. 0.54

作者: 冯雁

单位: 浙江大学

5-1

The function is to find the `K`-th smallest element in a list `A` of `N` elements. The function `BuildMaxHeap(H, K)` is to arrange elements `H[1]` ... `H[K]` into a max-heap. Please complete the following program.

```
ElementType FindKthSmallest ( int A[], int N, int K )
{   /* it is assumed that K<=N */
    ElementType *H;
    int i, next, child;

    H = (ElementType *)malloc((K+1)*sizeof(ElementType));
    for ( i=1; i<=K; i++ ) H[i] = A[i-1];
    BuildMaxHeap(H, K);

    for ( next=K; next<N; next++ ) {
        H[0] = A[next];
        if ( H[0] < H[1] ) {
            for ( i=1; i*2<=K; i=child ) {
                child = i*2;
                if ( child!=K && (3分) ) child++;
                if ( (3分) )
                    H[i] = H[child];
                else break;
            }
            H[i] = H[0];
        }
    }
    return H[1];
}
```

作者: 冯雁

单位: 浙江大学

时间限制: 400 ms

内存限制: 64 MB

5-2

The function is to find the `K`-th largest element in a list `A` of `N` elements. The initial function call is `Qselect(A, K, 0, N-1)`. Please complete the following program.

```
ElementType Qselect( ElementType A[], int K, int Left, int Right )
{
    ElementType Pivot = A[Left];
    int L = Left, R = Right+1;
    while (1) {
        while ( A[++L] > Pivot ) ;
        (3 分);
        if ( L < R ) Swap( &A[L], &A[R] );
        else break;
    }
    Swap( &A[Left], &A[R] );
    if ( K < (L-Left) )
        return Qselect(A, K, Left, R-1);
    else if ( K > (L-Left) )
        (3 分);
    else
        return Pivot;
}
```

作者: 陈越

单位: 浙江大学

时间限制: 400 ms

内存限制: 64 MB

6-1 Check Topological Order （8 分）

Write a program to test if a give sequence `Seq` is a topological order of a given graph `G`.

Format of functions:

```
bool IsTopSeq( Vertex Seq[], LGraph G );
```

where `LGraph` is defined as the following:

```
typedef struct AdjVNode *PtrToAdjVNode;
struct AdjVNode{
    Vertex AdjV;
    PtrToAdjVNode Next;
};

typedef struct Vnode{
    PtrToAdjVNode FirstEdge;
} AdjList[MaxVertexNum];

typedef struct GNode *PtrToGNode;
struct GNode{
    int N_v;
    int N_e;
    AdjList G;
};
typedef PtrToGNode LGraph;
```

The function `IsTopSeq` must return `true` if `Seq` does correspond to a topological order; otherwise return `false`.

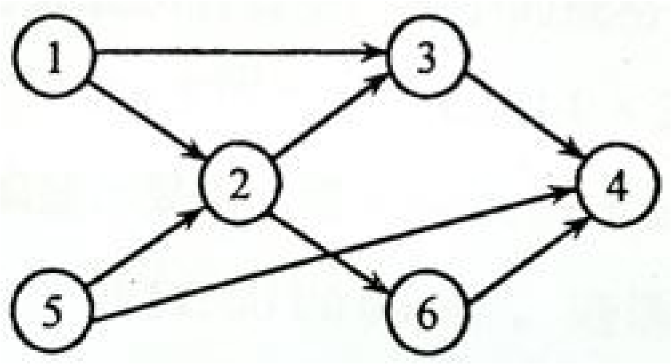
Note: Although the vertices are numbered from 1 to MaxVertexNum, they are **indexed from 0** in the LGraph structure.

Sample program of judge:

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

typedef enum {false, true} bool;
#define MaxVertexNum 10  /* maximum number of vertices */
typedef int Vertex;      /* vertices are numbered from 1 to MaxVertexNum */

typedef struct AdjVNode *PtrToAdjVNode;
```

```
struct AdjVNode{
    Vertex AdjV;
    PtrToAdjVNode Next;
};

typedef struct Vnode{
    PtrToAdjVNode FirstEdge;
} AdjList[MaxVertexNum];

typedef struct GNode *PtrToGNode;
struct GNode{
    int N_v;
    int N_e;
    AdjList G;
};
typedef PtrToGNode LGraph;

LGraph ReadG(); /* details omitted */

bool IsTopSeq( Vertex Seq[], LGraph G );

int main()
{
    int i, j, N;
    Vertex Seq[MaxVertexNum];
    LGraph G = ReadG();
    scanf("%d", &N);
    for (i=0; i<N; i++) {
        for (j=0; j<G->N_v; j++)
            scanf("%d", &Seq[j]);
        if ( IsTopSeq(Seq, G)==true ) printf("yes\n");
        else printf("no\n");
    }

    return 0;
}

/* Your function will be put here */
```

Sample Input (for the graph shown in the figure):



```
6 8
1 2
1 3
5 2
5 4
2 3
2 6
3 4
6 4
5
1 5 2 3 6 4
5 1 2 6 3 4
5 1 2 3 6 4
```

```
5 2 1 6 3 4

1 2 3 4 5 6
```

Sample Output:

```
yes

yes

yes

no

no
```

作者: 陈越

单位: 浙江大学

时间限制: 400 ms

内存限制: 64 MB

代码长度限制: 16 KB