判断题

- In an undirected acyclic graph G with 4 connected components, the number of edges will be 4 less than the number of vertices.
- Suppose that an array is used to store a circular queue, the value of `front` must be less than or equal to the value of `rear`.
- { 5, 2, 9, 18, 14, 16, 72 } cannot be the result after the second run of quick sort (assuming the pivot is chosen randomly).
- Quadratic probing isn't equivalent to double hashing with a secondary hash function of Has**$h2(k)=k$.
- Given a min-heap with unique keys, there are at least `h-1` keys less than any key in level `h` (the root level is `1`).
- In a tree of degree 3, we have $n2+n3 \geq n0$, where n**i is the number of degree i nodes for $0 \leq i \leq 3$.
- Let n be a non-negative integer representing the size of input. The time complexity of the following piece of code is $O(n)$.

  ```
  int func(int n){
      int sum = 0;
          for(int i = n; i > 0; i /= 2)
              for(int j = 0; j < i; j++)
                  sum++;
      return sum;
  }
  ```

- The FirstChild-NextSibling representation of a tree is unique.
- Suppose we are now sorting an initial array （8, 3, 9, 11, 2, 1, 4, 7, 5, 10, 6） using Shellsort. If the sorting result after the first run is (1, 3, 7, 5, 2, 6, 4, 9, 11, 10, 8) and that after the second run is (1, 2, 6, 4, 3, 7, 5, 8, 11, 10, 9), then the increments used for the two sortings are 5 and 2.
- Kruskal's minimum spanning tree algorithm implemented by disjoint set with union-by-rank strategy has $O(|E|log|E|)$ time complexity. Further optimization by introducing path compression improves it to $O(|E|\alpha(|E|,|V|))$ where α is the functional inverse of Ackermann's function.

单选题

A "full tree" of degree 3 (every non-leaf node has 3 children) has 31 nodes. Then its height h is at most . Note: h=0 for a single node tree.

A.11

B.10

C. cannot be determined

D. 9

The array representation of the disjoint sets is given by S[] with S[i] being initialized to be -1 for all i. Please list the resulting array elements after invoking: union(1,2), union(3,4), union(3,5), union(1,7), union(3,6), union(8,9), union(1,8), union(1,3), and union(9,10). Assume that union-by-size and find-with-path-compression, and the elements are numbered from 1 to 10.

A. {-10, 1, 1, 3, 3, 3, 1, 1, 1, 1}

B. {-3, 1, -4, 3, 3, 3, 1, -2, 8, –1}

C. {-9, 1, 1, 3, 3, 3, 1, 1, 1, 1}

D. {-9, 1, 1, 3, 3, 3, 1, 1, 8, –1}

Let *n* be a non-negative integer representing the size of input. The time complexity of the following piece of code is:

```
void func(int n){
  int i = 0;

  while(i * i * i <= n)
    i++;
}
```

A. *O*($n^{\frac{1}{3}}$)

B. *O*(n^3)

C. *O*(log_3*n*)

D. *O*(*n*)

For a given binary tree, if its post-order traversal sequence is { 3, 4, 1, 6, 2, 5 }, pre-order traversal sequence is { 5, 3, 2, 1, 4, 6 }, then its in-order traversal sequence must be:

A. 3, 5, 1, 4, 2, 6

B. cannot be determined

C. 5, 3, 2, 1, 6, 4

D. 3, 5, 4, 1, 2, 6

Push 4 characters `oppo` onto a stack. In how many different ways that we can pop these characters and still obtain `oppo`?

A. 6

B. 4

C. 3

D. 5

A binary tree with 2825 nodes must have a height . (Assume the height of a single node tree is 0.)

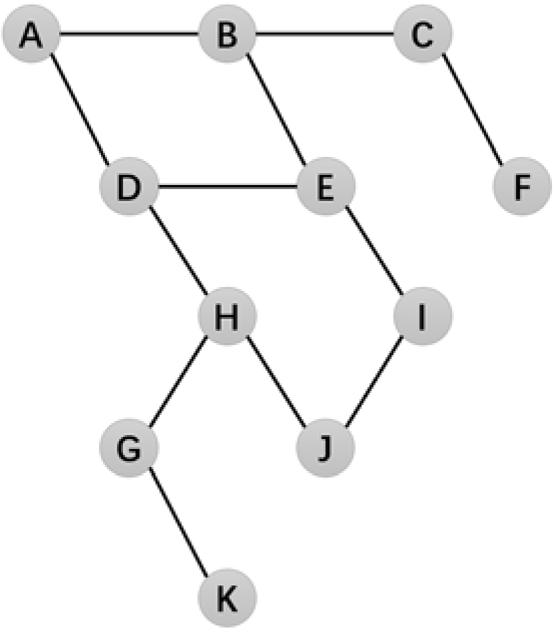
A. in the range [11, 2824]

B. in the range [12, 1412]

C. in the range [11, 2825]

D. in the range [10, 2824]

How many articulation points are there in the undirected graph below?



A. 4

B. 2

C. 3

D. 5

Which one of the following statement is FALSE about a tree?

A. Sibling nodes share the same parent.

B. When converting a tree into a binary tree using FirstChild-NextSibling representation, the root of the binary tree must be of degree 1.

C. A nonempty tree must have a root, which has zero or more nonempty subtrees.

D. The degree of a tree is equal to the height of the tree.

To obtain an ascending sequence, which of the following sorting methods gives {16, 17, 20, 7, 8, 10, 28, 5, 3} after 2 runs?

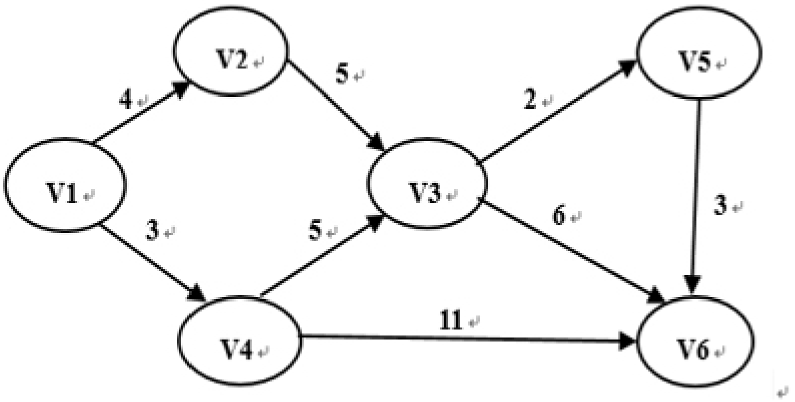
A. quick sort

B. insertion sort

C. selection sort

D. merge sort

For the given directed weighted graph of 6 vertices, which one of the following sets of vertices satisfies that the earliest completion time of every member is equal to its latest completion time?



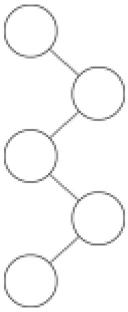
A. V1 V2 V3 V6

B. V1 V2 V3 V5 V6

C. V1 V4 V3 V6

D. V1 V4 V6

Given a binary search tree as shown below, which one of the following insertion orders is valid?



A.

```
11 15 12 14 13
```

B.

```
11 15 13 14 12
```

C.

```
11 12 13 14 15
```

D.

```
12 15 11 14 13
```

When Dijkstra's algorithm is used to find the shortest paths from *v*1 to every other vertices in an undirected graph G, a distance set dist[*v*] is maintained for every vertex *v*, as the shortest distance from *v*1 to *v*, passing through only the vertices whose shortest path to *v*1 has been determined.

Suppose that the destinations are obtained in the order of { *v*2, *v*3, ⋯, vn }. If there is a vertex *w* in G so that dist[*w*] is decreased during every iteration (except the last one) of the algorithm, how many statements of the following are **correct**?

(1) *w* is v_n.

(2) *w* is adjacent to every other vertices.

(3) (*v*1, *w*) (if it exists) is the longest edge among all the edges that are adjacent to *w*.

A. 3

B. 0

C. 2

D. 1

Which of the following is the sufficient and necessary condition that an undirected connected graph has Euler circuit?

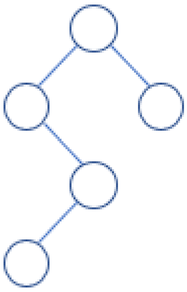
A. At most 2 vertices have odd degrees

B. All the vertices have even degrees

C. Exactly 2 vertices have odd degrees

D. At most 1 vertex has an odd degree

The inorder traversal sequence of the given tree is { 3, 1, 2, 5, 4 }. Which of the following statements is FALSE?



A. 3 is the left child of 1

B. 5 is the root

C. 1 is the left child of 2

D. 4 is a leaf node

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

A. 0.46

B. 0.31

C. 0.23

D. 0.38

Let us convert a general tree T into a binary tree BT. Suppose that there are $n0$ leaf nodes in T and $m0$ leaf nodes in BT. Which of the following relationship between $n0$ and $m0$ is true?

A. $n0 \leq m0$

B. $n0 \geq m0$

C. cannot be determined

D. $n0 = m0$

Which sorting method can find the final position of at least one element within the sorted list after each run?

1. selection sort
2. quick sort
3. Shell sort
4. merge sort
5. heap sort

A. 1,3,4

B. 3,4,5

C. 1,2,5

D. 2,3,4

If a complete binary tree with `2000` nodes is represented by an array (root index is `1`), then the index of the lowest (deepest) common ancestor of the node at index `910` and the node at index `480` is .

A. 30

B. 14

C. 7

D. 56

Given a hash table of size 13 and the hash function $h(x)=x\%13$. double hashing is used to solve collisions with $h(x)=(h(x)+ih2(x))\%13$ for $i=1,2,\ldots,12$, where $h2(x)=(x\%11)+1$. After filling in the hash table one by one with input sequence {92,81,58,21,57,45,161,38,117}, which number is placed in the position of index 6?

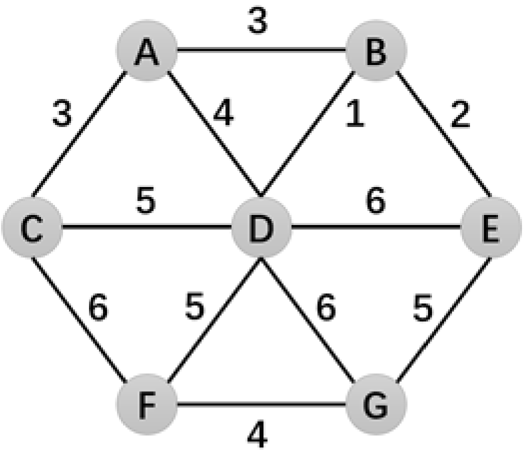
A. 58

B. 57

C. 81

D. 21

Given a weighted undirected graph as shown below, with a given starting vertex. Which of the following is a possible sequence of edges collected by Prim's minimum spanning tree algorithm?



A. Starting from A: (A,B) -> (B,D) -> (B,E) -> (A,C) -> (D,F) -> (F,G)

B. Starting from B: (B,D) -> (B,E) -> (A,B) -> (A,C) -> (F,G) -> (D,F)

C. Starting from A: (A,C) -> (C,D) -> (D,F) -> (F,G) -> (E,G) -> (B,E)

D. Starting from B: (B,D) -> (A,D) -> (A,B) -> (B,E) -> (E,G) -> (F,G)

程序填空题

The topological sort with a stack

The following program implements the topological sort algorithm with a **stack**. Please fill in the blanks.

```
void Topsort(int a[NUM][NUM], int TopNum[NUM])
// a[NUM][NUM] is adjacency matrix of the graph with NUM vertices
// TopNum[NUM] stores the topological orders
{   int S[NUM],  Indegree[NUM];        //S[NUM] is a stack
    int  Counter = 0, top, n, i, j;
    int V;
    top= -1;
    n=NUM;
    for (j=0; j<n; j++) {
        Indegree[j]=0;
        for (i=0; i<n; i++)
            if ( (1) ) Indegree[j]++;
        if ( Indegree[j] == 0 )   S[++top]=j;
    }
    while (top>=0) {
        V = S[top--];
        TopNum[ V ] = ++ Counter;     /* assign next */
        for (j=0; j<n; j++)
           if ( a[V][j]!=0)
              if ( (2) == 0 )  S[++top]=j;
    }  /* end-while */
    if ( Counter!=n )  printf( "Graph has a cycle" );
}
```

Heapsort

The function is to sort the array `A` in descending order. The function `Swap(&A[0], &A[i])` is to exchange `A[0]` and `A[i]`. Please complete the following program.

```
#define leftchild(i) (2*(i)+1)
void PercDown(ElementType A[], int i, int N) {
    int child;
    ElementType tmp;
    for (tmp = A[i]; leftchild(i) < N; i = child) {
        child = leftchild(i);
        if ( (1) )
            child++;
        if (tmp > A[child])
            A[i] = A[child];
        else
            break;
    }
    A[i] = tmp;
}
void Heapsort(ElementType A[], int N) {
    int i;
    for (i = N / 2; i >= 0; i--)
        PercDown(A, i, N);
    for (i = N - 1; i > 0; i--) {
        Swap(&A[0], &A[i]);
         (2) ;
    }
}
```

函数题

Print the left subtree of X in BST

Given a binary search tree `T`, you are supposed to output all the elements in the left subtree of `X`, in *ascending order*.

Format of functions:

```
void print_left_subtree( Tree T, int X );
```

where `Tree` is defined as:

```
typedef struct TreeNode *Tree;
struct TreeNode {
    int element;
    Tree left;
    Tree right;
};
```

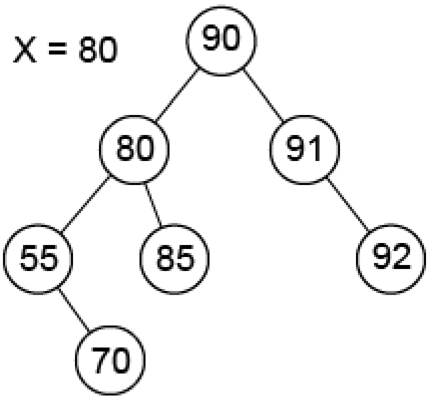
Sample program of judge:

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

typedef struct TreeNode *Tree;
struct TreeNode {
    int element;
    Tree left;
    Tree right;
};

Tree build_tree(); /* details omitted */

void print_left_subtree(Tree T, int X);

int main()
{
    Tree T;
    int X;

    T = build_tree();
    scanf("%d", &X);
    print_left_subtree(T, X);

    return 0;
}

/* Your function(s) will be put here */
```

Sample Input

```
7
90 80 91 55 70 85 92
80
```

Sample Output:



For the sample above, your program should output:

```
55 70 end
```

For an empy tree, simply output:

```
empty tree
```

In case `X` is not found in `T`, output:

```
x not found
```

where the letter "x" is NOT the value of `X`.

答案

TFFTT FTFFF

BAABC AADBA AABAD BCCAA

`a[i][j]!=0` `(--Indegree[j])`

`((child+1)<N) && (A[child]>A[child+1])` `PercDown(A, 0, i)`

```
void dfs(Tree T)
{
    if(T==NULL) return;
    dfs(T->left);printf("%d ",T->element);dfs(T->right);
}

void print_left_subtree( Tree T, int X )
{
    if(T==NULL){printf("empty tree\n");return;}
    while(T!=NULL && T->element!=X){
        if(T->element>X) T = T->left;
        else T = T->right;
    }
    if(T==NULL){printf("x not found\n");return;}
    T=T->left;//if(T==NULL){printf("empty tree\n");return;}
    dfs(T);printf("end");
}
```