浙江大学2024-2025学年春夏学期 《数据结构基础》课程期末考试试卷

判断题

1. To find a minimum spanning tree in a sparse graph, Prim's algorithm is more suitable than Kruskal's.
2. In an undirected graph, a vertex v is an articulation point iff there exists a pair of vertices u and w (both different from v) so that every path between u and w passes through v.
3. Given a min-heap with unique keys, if $i < j$, then the element p in level i must be smaller than q in level j.
4. Suppose that each single node in a disjoint set initially has height 0, the height of the resulting disjoint-set tree formed by performing union-by-height on n nodes may exceed $\lfloor \log_2 n \rfloor$.
5. If quadratic probing is used in hashing, then a new element can always be inserted if the hash table is at least half empty.
6. After performing one pass of radix sort on {05, 46, 13, 55, 94, 17, 42}, the result could be {05, 13, 17, 42, 46, 55, 94}.
7. During the insertion sort process, the order of magnitude of the number of comparisons is independent of the initial state of the sequence.
8. $114514^N = O(N!)$
9. If a complete binary tree has exactly n nodes in each of its last two levels, then the total number of nodes in the tree must be 3n-1.
10. If the five integers {1,2,3,4,5} are inserted into an initially empty binary search tree in any order starting with 3, the resulting trees will all have the same height.

单选题

1. Given a tree of degree 3, if there are 49 nodes of degree 2, and 25 nodes of degree 3, then how many leaf nodes are there?
    - A. 101
    - B. 100
    - C. 150
    - D. 50
2. Which of the following statements is correct regarding a binary search tree storing unique integers?
    - A. The integers 12 and 13 may be at the same level.

- B. If 12 and 14 are at the same level, then 13 must be their parent node.
- C. If 12 and 14 are at the same level, then 13 must not be their parent node.
- D. None.

3. For a binary tree, if its post-order travel sequence is { 11, 13, 12, 16, 15, 17, 14 }, and its in-order travel sequence is { 11, 12, 13, 14, 15, 16, 17 }, then which of the following statement is TRUE?
   - A. 16 is the parent of 15.
   - B. 11 is the parent of 12.
   - C. This is a complete binary tree.
   - D. This is a binary search tree.
4. In a doubly linked circular list, how many pointers must be updated to insert q after p?
   - A. 1
   - B. 2
   - C. 3
   - D. 4
5. Design an algorithm to determine whether the parentheses () in an expression are properly matched. Which data structure is the most suitable for this task?
   - A. Linked linear list
   - B. Queue
   - C. Stack
   - D. Binary Tree
6. Suppose that the level-order traversal sequence of a min-heap is { 5, 18, 29, 23, 64, 47, 36}. After calling DeleteMin twice, what will be the post-order traversal of the resulting heap after it has been adjusted?
   - A. 47, 36, 64, 23, 29
   - B. 47, 64, 36, 23, 29
   - C. 47, 64, 29, 36, 23
   - D. 47, 64, 36, 29, 23
7. It is known that a 3-heap is a heap whose nodes have 3 children. In a max-3-heap with 2025 elements, which one of the following indices is NOT a possible position for the minimum key?
   - A. 675
   - B. 676
   - C. 1012
   - D. 1013

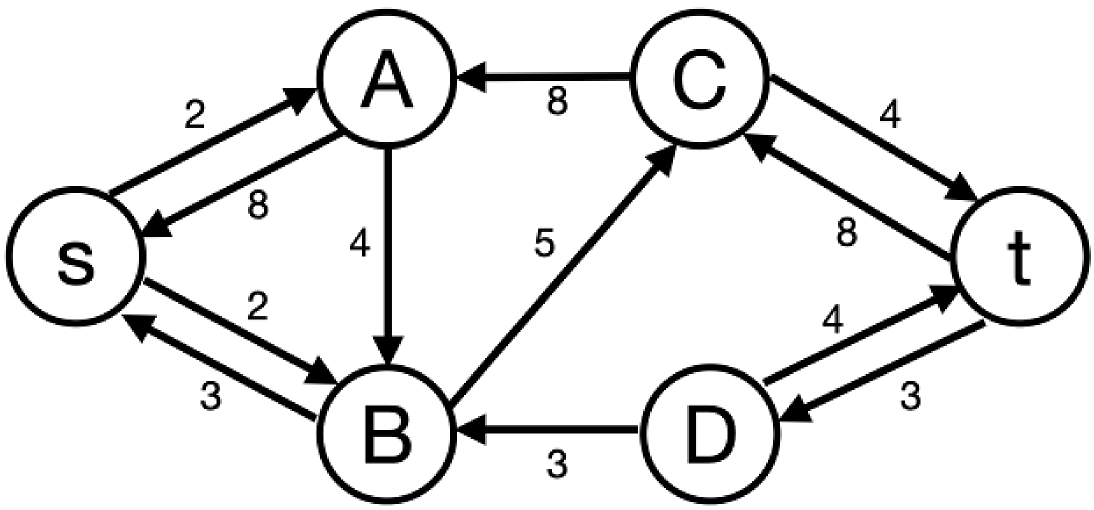
8. For an in-order threaded binary tree, if the post-order and in-order traversal sequences are FDEBGCA and FDBEACG, respectively. Which pair of nodes' right links are both threads?
    - A. C and D.
    - B. B and G.
    - C. C and F.
    - D. E and G.
9. Consider the following operations performed on an initially empty Binary Search Tree (BST):Insert: 100, 50, 150, 25, 75, 125, 175Delete: 100 (replaced with its in-order successor)Insert: 110After these operations, which node will be the in-order successor of 75 in the final BST?
    - A. 125.
    - B. 110.
    - C. 50.
    - D. The in-order successor does not exist for 75.
10. The array representation of the disjoint sets is given by { 4, 5, 4, 5, -6, 9, -2, 7, -3, 9, 2 }. Keep in mind that the elements are numbered from 1 to 11, and the depth of the root is 0. After invoking Union(Find(6), Find(8)), Union(Find(10), Find(3)) with union-by-size and path compression, what is the maximum depth of all the nodes?
    - A. 1
    - B. 2
    - C. 3
    - D. 4
11. Which of the following statements about hashing is incorrect?
    - A. In hashing with quadratic probing to resolve collisions, a new element can definitely be inserted if the table size is 14 and 7 cells are occupied.
    - B. If an insertion into the hash table fails, rehashing is necessary, and the new table size is about twice the original size.
    - C. If in double hashing $\mathrm{Hash}_2(k) = k^2$, then double hashing effectively becomes quadratic probing.
    - D. the expected number of probes for insertions is equal to unsuccessful searches in linear probing method.
12. Insert {16,29,10,24,27,11,31} one by one into an initially empty hash table of size 11 with the hash function $H(Key) = Key\%11$, and linear probing is used to resolve collisions. After Insertion, what is the expected number of probes for an unsuccessful search?
    - A. $\frac{20}{11}$
    - B. $\frac{24}{11}$

- C. $\frac{22}{11}$
- D. $\frac{23}{11}$

13. To compute the shortest path in a weighted graph, Dijkstra's algorithm using a min-priority queue is a popular choice. For this algorithm, what is the total time spent on running the relaxation step "decrease the distance of each vertex adjacent from the current visited one"?

    - A. $O(|E|)$
    - B. $O(|V| * |E|)$
    - C. $O(|V| * \log |V|)$
    - D. $O(|E| * \log |V|)$

14. We are executing an algorithm to find the maximum flow of a given network. Currently, we have found several augmenting paths to send 11 flow from s to t, and we show the updated residual graph in the following figure.



So what is the maximum flow we can send from s to t?

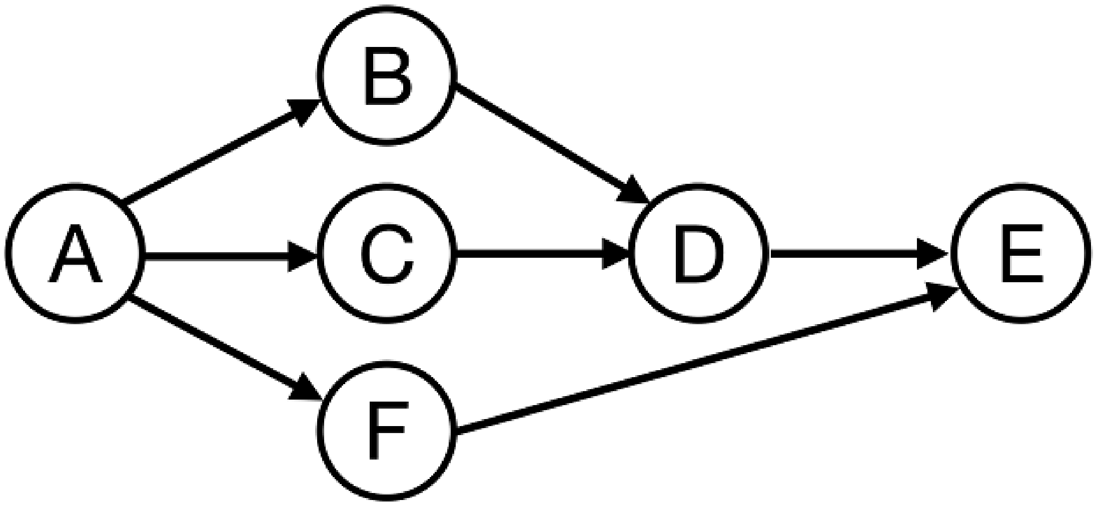
```
- A. 11

- B. 13

- C. 15

- D. 19
```

1. Given a weakly-connected directed graph, what condition ensures that an Eulerian Circuit exists?

    - A. All vertices have equal in-degree and out-degree.
    - B. Exactly one vertex has out-degree = in-degree + 1, one vertex has in-degree = out-degree + 1, and all the rest vertices have equal in-degree and out-degree.
    - C. Exactly two vertices have odd total-degree, and all the rest vertices have even total-degree.

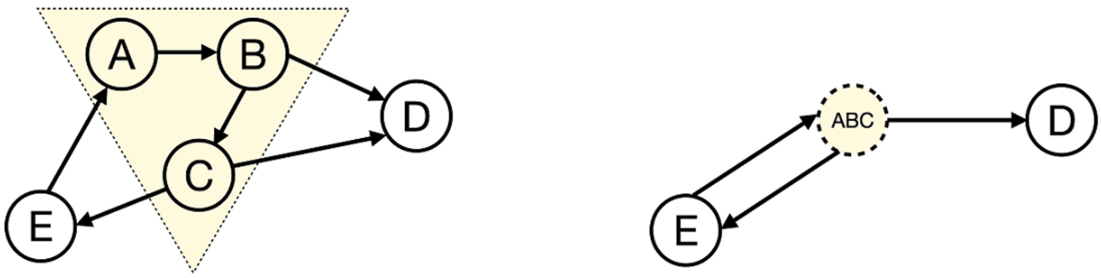
- D. All vertices have even total-degree.

2. How many valid topological sorts does the following graph have?



- A. 7
- B. 8
- C. 9
- D. 10

3. The following figure illustrates how to contract a subgraph (shown in dashed line) into a vertex.



Now given a directed graph with many strongly connected components, if we contract each component into a single vertex, which property does the resulting graph MUST have?

- A. a complete graph
- B. a directed acyclic graph
- C. a disconnected graph
- D. a binary tree

4. Using the direct insertion sort algorithm to sort the following four lists in ascending order, the one with the fewest number of comparisons is:

- A. 67,13,20,55,40,27,6,31
- B. 6,13,27,20,40,31,55,67
- C. 13,20,6,27,31,67,55,40
- D. 55,31,40,27,6,13,67,20

5. Given that the key sequence {10, 12, 17, 22, 28, 23, 19, 24} is a min-heap, after inserting the key 5 and adjusting accordingly, the resulting min-heap is:
    - A. {5,17,10,12,28,23,19,24,22}
    - B. {5,10,17,22,23,19,24,12,28}
    - C. {5,12,17,10,23,19,24,28,22}
    - D. {5,10,17,12,28,23,19,24,22}
6. Which of the following options is not a possible result after the second pass of quicksort?
    - A. 12,16,44,18,50,52,58
    - B. 12,52,44,50,18,16,58
    - C. 16,12,44,18,52,50,58
    - D. 18,12,16,44,52,50,58

程序填空题

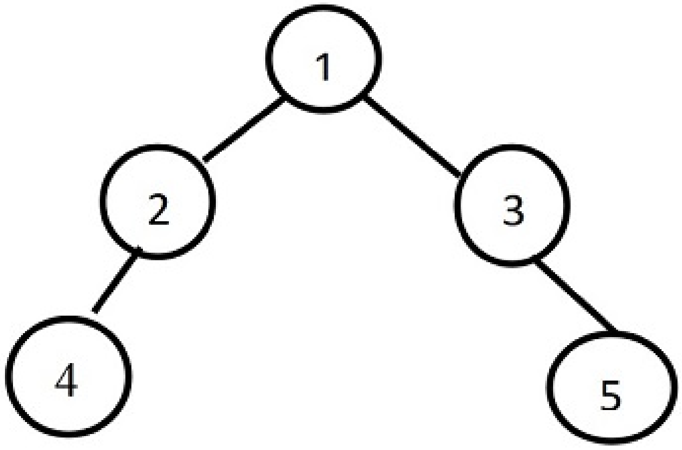
1. Given the head of a singly linked list, swap every two adjacent nodes if the value of the first node is greater than the value of the second node and the value of the first node is even. If the number of nodes is odd, the last node remains unchanged. Complete the two blanks in the code below.

```
#include <stdio.h>
#include <stdlib.h>
struct ListNode {
    int val;
    struct ListNode *next;
};
struct ListNode* swapIfGreaterAndEven(struct ListNode* head) {
    struct ListNode dummy;
    dummy.next = head;
    struct ListNode *prev = &dummy;

    while (prev->next != NULL && prev->next->next != NULL) {
        struct ListNode *first = prev->next;
        struct ListNode *second = first->next;

        if ( ____（3 分） ) {
            first->next = ____（3 分）;
            second->next = first;

            prev->next = second;
            prev = first;
        } else {
            prev = second;
        }
    }
    return dummy.next;
}
```

2. In a binary tree represented as a complete binary tree, find the first node with a right child encountered during an in-order traversal. If no such node exists, return -1; otherwise, return that node's value.

For example: a binary tree stored in this format (as illustrated in the figure) has its root node at array index 1, with missing nodes represented by 0 in the array.

Please complete the function int FirstHaveRight(int a[], int n), where parameter n is the index of the last element in array a (e.g., n = 7 in the provided example). All elements after the last element of array a are initialized to 0. The algorithm in this function is similar to iterative implementation of in-order traversal.



数组 a	--	1	2	3	4	0	0	5	
下标	0	1	2	3	4	5	6	7	

```
int FirstHaveRight(int a[],int n)
{  int stack[MAXSIZE];
   int top=-1, tree=1, treeRight;

   for ( ; ; ) {
     for ( ; tree<=n; tree = 2*tree){
         if ( ____（3 分） ) stack[++top]=tree;
         else break;
     }
     if (top==-1) return -1;
     else tree = stack[top--];
     treeRight= ____（3 分）;
     if (a[treeRight]!=0) break;
     else tree = treeRight;
   }
   return a[tree];
}
```

函数题：Finding Independent Set

题目描述

Given a simple undirected graph $G = (V, E)$. An **independent set** of G is a set $S \subseteq V$ such that no two members of S are connected by an edge in E. Finding the maximum independent set of G is an NP-hard problem. Here you are supposed to implement a greedy hueristic to find a near-maximum independent set.

The algorithm works in the following way:

1. Collect any one unvisited vertex v into S.
2. Delete all the vertices (and all the edges incident on them) that are adjacent to v from G.
3. Repeat steps 1 and 2 until there is no unvisited vertex left in G.

In order to obtain the unique solution, when there are many options in step 1, you must always choose the vertex with the smallest index.

Format of functions:

```
int Independent_Set( MGraph G, int S[] );
```

Here `MGraph` is defined as the following:

```
typedef struct GNode *PtrToGNode;
struct GNode{
    int Nv;  /* number of vertices */
    int Ne;  /* number of edges    */
    int G[MaxVertexNum][MaxVertexNum]; /* adjacency matrix */
};
typedef PtrToGNode MGraph;
```

where `MaxVertexNum` is a constant defined by the judge.

The indices of the vertices in the resulting independent set is stored in the array `int S[]`. Note that the vertices are indexed from 1 to the number of vertices `Nv`. However, the indices of the adjacency matrix `G` starts from 0.

The function `Independent_Set` is supposed to follow the above algorithm to produce an independent set, save the results in `S`, and return the number of vertices in `S`.

Sample program of judge:

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

#define MaxVertexNum 1000

typedef struct GNode *PtrToGNode;
struct GNode{
    int Nv;  /* number of vertices */
    int Ne;  /* number of edges    */
    int G[MaxVertexNum][MaxVertexNum]; /* adjacency matrix */
};
typedef PtrToGNode MGraph;

MGraph Build_Graph(); /* implemented by judge */
int Independent_Set( MGraph G, int S[] );

int main()
{
    int i, n, S[MaxVertexNum];
    MGraph G;

    G = Build_Graph();
    n = Independent_Set(G, S);
    printf("%d", S[0]);
    for (i=1; i<n; i++) printf(" %d", S[i]);

    return 0;
}
/* Your function(s) will be put here */
```

Sample Input:

```
8 7
1 5
5 4
4 2
2 3
3 6
6 1
6 2
```

Sample Output:

```
1 2 7 8
```

Note:

This method can help us find an independent set, but not necessarily a **maximum** independent set. For example, the maximum solution of the sample graph is {1, 3, 4, 7, 8}.

题目要求

代码长度限制：16 KB

运行时间限制：400 ms

内存限制：65536 KB