94

Data Structure(B 卷)

Midterm Exam. 2

2022/11/24

學號: 410470575 姓名: 王重勛

總分 105 分 最高以 100 分計

1. Given the *adjacency matrixs* with cost weights of an undirected graph G2.

$$G2 = \begin{bmatrix} 0 & 6 & 3 & 2 & 0 \\ 6 & 0 & 5 & 0 & 7 \\ 3 & 5 & 0 & 4 & 1 \\ 2 & 0 & 4 & 0 & 0 \\ 0 & 7 & 1 & 0 & 0 \end{bmatrix}$$

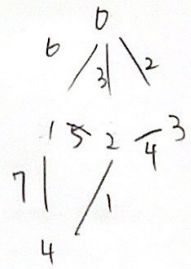
15

(A) Apply Kruskal's Algorithm to find the Minimum Spanning Tree of this graph. Please give number on the selected edge to describe the order that the edges are selected. (7 分)

Ans:

			cost	selected order
E1	0	1	6	
E2	0	2	3	3
E3	0	3	2	2
E4	1	2	5	4
E5	1	4	7	
E6	2	3	4	
E7	2	4	1	1

(B) Apply Prim's algorithm to find the Minimum cost Spanning Tree of this graph **starting from vertex 0**. Please give number on the selected edge to describe the order that the edges are selected. (8 分)



Ans:

			cost	selected order
E1	0	1	6	
E2	0	2	3	2
E3	0	3	2	1
E4	1	2	5	4
E5	1	4	7	
E6	2	3	4	
E7	2	4	1	3

2.

(A) Please complete the following code to insert an element *item* into a minimum heap stored in an array heap[] with current size *n. You can assume the array still has enough space for the insertion. (6 分)

13

```
void insert_minimum_heap(element item, int *n){
    int i;

    i = ++(*n);
    while (____<1>_____&&_____<2>_____){
        heap[i] = heap[i/2];
        i/= 2;
    }
    _____<3>_____ = item;
}
```

Ans:

<1>	i >= 1
<2>	heap[i/2].~~key~~ > item.~~key~~
<3>	heap[i]

(B) Please complete the following code to delete the minimum element from a minimum heap stored in an array heap[] with current size *n. (8 分)

```
element delete_minimum_heap(int *n){
    int parent, child;
    element item, temp;
    if (*n == 0) {
        fprint(stderr, "The heap is empty\n");
        exit(1);
    }
    item = heap[1];
    temp = ____<1>_____;
    parent = 1;
    child = 2;
    while (child <= *n) {
        if (child < *n) && (heap[child].key ____<2>_____)
            child++;
        heap[parent] = _____<3>_____;
        parent = child;
        child = _____<4>_____;
    }
    heap[parent] = temp;
    return item;
}
```

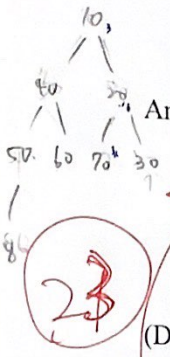
Ans:

-1

<1>	heap[~~*~~n] heap[*n]
<2>	> heap[child+1].key
<3>	heap[child]
<4>	parent * 2

(C) Suppose that the content in heap[] is shown as below, please show the result after inserting 20 and 40 one by one. (4 分)

Index i	1	2	3	4	5	6	7	8
Heap[i]	10	50	30	80	60	70		



Ans:

Index i	1	2	3	4	5	6	7	8
Heap[i]	10	50	20	80	60	70	30	

10 40 20 50 60 70 30 80

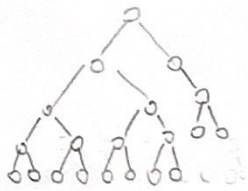
(D) Please show the result after deleting 2 elements from the result of (C). (4 分)

Ans:

Index i	1	2	3	4	5	6	7	8
Heap[i]	10	40	20	50	60	70	30	80

30 40 70 50 60 80

3. Suppose that the number of nodes of a binary tree is 20 and the number of leave nodes is 10. (9 分)



What is the number of nodes in the binary tree with degree 2? Ans: 9

Whats is the number of nodes in the binary tree with degree 1? Ans: 1

What is the number of edges in the binary tree? Ans: 19

4. Given a binary search tree, which binary tree travelsal can be used to output the data in sorted order? (5 分)

(A) Level order traversal (B) Preorder traversal (C) Inorder traversal (D) postorder traversal

Ans: C

5. Given a binary expression tree, which binary tree travelsal can be used to evaluate the result of expression? (5 分)

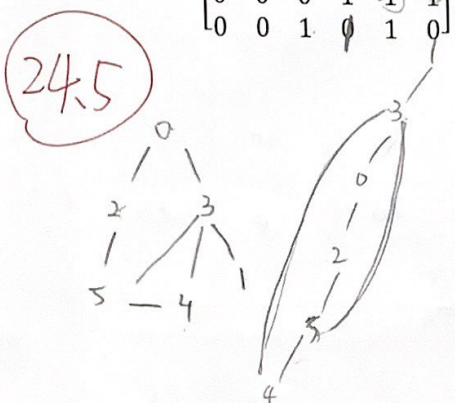
(A) Level order traversal (B) Preorder traversal (C) Inorder traversal (D) postorder traversal

Ans: D

6. Given the *adjacency matrixs* of an undirected graph G.

甲、 Please represent the graph by an *adjacency multilist*. (10 分)

$$G = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$



Ans:

Head Nodes

[0]	E_1
[1]	E_3
[2]	E_1
[3]	E_2
[4]	E_5
[5]	E_4

E1	0	2	E_2	E_4
E2	0	3	NULL	E_3
E3	1	3	Null	E_5
E4	2	5	Null	E_6
E5	3	4	E_6	E_7
E6	3	5	Null	E_7
E7	4	5	Null	Null

-0.5

(B) Suppose the graph is visited in DFS. Besides, the neighbers of a node are visited in alphanumerical order if the node has more than one neighbor. Please write the **DFN** (the number indicating the depth first search order that the node is visited) of each node when **starting from node 1**. (5 分)

Ans:

Index i	0	1	2	3	4	5
DFN[i]	3	1	4	2	6	5

(C) Please write the *Low* value of each node (the lowest *dfn* that can be reached from a node passing a path of descendants followed by at most one back edge) when **starting from node 1**. (5 分)

Ans:

Index i	0	1	2	3	4	5
Low[i]	2	1	2	2	2	2

(D) Please write the articulation points of the graph G. (5 分)

Ans: 3

7. The following statement is used to define a node in a threaded binary tree, where the leftThread of a node points to the previous inorder node and the rightThread of the node points to the next inorder node. (15 分)

```
typedef struct threadedTree *threadPointer;
typedef struct threadedTree {
   boolean leftThread, rightThread;
   threadPointer leftChild, rightChild;
   char data;
   };
```

(A) Please fill in the code to complete the function for finding the pointer pointing to the successor of the given node ***tree***.

11.5

```
threaded_pointer insucc(threadPointer tree){
   thread_pointer temp;
   temp = tree->rightChild;

   if (____<1>_____)

        while (_____<2>_____)
             temp = temp->leftChild;
   return temp;
}
```

Ans:

<1>	!tree->rightThread
<2>	temp->leftChild

temp

-2.5

(B) Please fill in the code to complete the function for inserting ***r*** as the right child of ***s*** in the threaded binary tree.

```
void insert_right(threadPointer s, threadPointer r){
   threaded_pointer temp;
   r->rightChild = ____<1>_____;
   r->rightThread = s->rightThread;
   r->leftChild = s;
   r->leftThread = ____<2>_____;
   s->rightChild = r;
   s->rightThread = FALSE;
   If(_____<3>_____){
        temp = insucc(r);
        _____<4>_____ = r;
   }
}
```

Ans:

<1>	s->rightChild
<2>	true
<3>	!r->rightThread / r->rightthread
<4>	temp->leftChild

8. Please give the time complexity in **worst case** when using the following data structure to perform the following operations. It is assumed that the data structure contains n data and the corresponding binary search tree with height h. (9 分)

	Insert a data	Delete the maximum data	Search any data
Unordered array	~~O(n)~~ O(1)	O(n)	O(n)
A sorted linked list	~~O(1)~~ O(n)	O(1)	O(n)
A binary search tree	O(h)	O(h)	O(h)

7

-2