《数据结构及其算法》往年题答案

陈镜舟

08-09第一学期

一

1. (2/3)^n^,2^100^,log~~2~~n,n,nlog~~2~~n,n^3/2^,n^n^
2. [(n+1)/2]
3. 123,132,213,231,321
4. (n+1)/2
5. 2h-1
6. 统计邻接矩阵第i-1列中非0元素的个数
7. 37/12
8. 47
9. C
10. D
11. A
12. D
13. D
14. C
15. A
16. C
17. B
18. B

二

1. (1)3,14,6,17,32,9,31,7,12,24,20 (2)3,14,17,32,9,31,6,12,7,24,20 (3)20,31,32,12,3,14,24,6,17,7,9

2. 地址依次为：100,501,1629

3. (1) (2)(a+(b-c/((d+e)* f))* g)-h (3)8

4. (1)V1,V2,V5,V8,V6,V7,V3,V4 (2)V8,V5,V7,V6,V2,V3,V4,V1 (3)

事件	V1	V2	V3	V4	V5	V6	V7	V8
v~~e~~	0	6	38	30	89	51	75	97
v~~l~~	0	27	38	30	89	51	75	97

活动	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13
e(i)	0	0	0	6	30	6	6	38	30	30	51	51	89
l(i)	21	33	0	27	30	46	45	38	33	63	51	51	89

关键路径1：V1->V4->V3->V6->V7->V8 关键路径2：V1->V4->V3->V6->V5->V8

活动提前：a3、a5、a8

5. (1) (2)ASL=2.6 (3)

三

2. 题目描述有点模糊，故对其略作修改和补充：a.找到了返回指向该结点的指针，找不到返回nullptr b.已知链表的最后一个结点（即参数high）由于空间复杂度O(1)的限制，不能使用递归

```
//(1)
struct linkedListNode {
int data;
linkedListNode* next;
};
linkedListNode* BiSearch(linkedListNode* pHead, int key, linkedListNode* low,
linkedListNode* high) {
    linkedListNode* mid, * fast;
    mid = low;
    fast = low;
    while (high->next != low) {
        //快慢指针法找出中点的直接前驱，之所以要找直接前驱是因为方便得到(从数组视角
看)mid-1
        while (fast != high) {
            fast = fast->next;
            if (fast != high) {
                fast = fast->next;
                mid = mid->next;
            }
        }

        if (mid->next->data == key)return mid->next;
        else if (mid->next->data < key)low = mid->next->next;
        else high = mid;
    }
    return nullptr;
}
//(2)
//对(1)的算法略作修改即为(2)答案
void insert(linkedListNode* pHead, int key, linkedListNode* low, linkedListNode*
high) {
    linkedListNode* mid, * fast;
    mid = low;
    fast = low;
    while (high->next != low) {
        //快慢指针法找出中点的直接前驱，之所以要找直接前驱是因为方便得到(从数组视角
看)mid-1
        while (fast != high) {
            fast = fast->next;
            if (fast != high) {
                fast = fast->next;
                mid = mid->next;
            }
        }

        if (mid->next->data == key)return;
        else if (mid->next->data < key)low = mid->next->next;
        else high = mid;
    }
    //查找失败时，high指向目标插入位置的前一个结点
    linkedListNode* ptr = new linkedListNode;
    ptr->data = key;
    ptr->next = high->next;
    high->next = ptr;
}
```

3.
   ```
   #include<queue>
   using namespace std;
   bool isCompleteBT(node* T) {
       queue<node*> Q;
       node* p = T;
       while (p) {
           Q.push(p->lchild);
           Q.push(p->rchild);
           p = Q.front();
           Q.pop();
       }
       while (!Q.empty()) {
           p = Q.front();
           Q.pop();
           if (p)return false;
       }
       return true;
   }
   ```

09-10第一学期

一

1. 对
2. 对
3. 错
4. 对
5. 错
6. 错
7. 对
8. 对
9. 对（存疑）
10. 错（存疑）

二

1. C
2. A
3. 不知道
4. D
5. A
6. C
7. C
8. 71
9. 4
10. 7;右;10;447

三

1. 优先选取序号小的结点，以边两端的结点对表示边 Kruskal: (v4,v6) (v2,v5) (v4,v7) (v3,v7) (v1,v2) (v4,v5) Prim: (v1,v2) (v2,v5) (v5,v4) (v4,v6) (v4,v7) (v7,v3)

2. (1)

	1	2	3	4	5	6	7	8	9	10
v~~e~~	0	5	6	12	15	16	16	19	21	23
v~~l~~	0	9	6	12	15	19	16	19	21	23
(2)										
23										
(3)										
a2,a4,a6,a8,a9,a10,a11,a13										
(4)										
a2,a4,a6,a13										

3. (1)BST略，ASL=18/7 (2)8

4. 过程略，最终结果：

	S	D		P
v~~1~~	1	0		
v~~2~~	1	25		1,5,2
v~~3~~	1	43		1,5,2,3
v~~4~~	1	55		1,5,2,3,4
v~~5~~	1	10		1,5
故v~~1~~到v~~4~~最短路径:v~~1~~->v~~5~~->v~~2~~->v~~3~~->v~~4~~，长度=55				

5. (1)返回值为4 (2)求树深

四

1.
```
void nodeNum(node* root, int depth, int L, int& sum) {
    if (!root)return;
    if (depth == L)sum++;
    nodeNum(root->left, depth + 1, L, sum);
    nodeNum(root->right, depth + 1, L, sum);
}

int nodeNum(node* root, int L) {
    int sum = 0;
    nodeNum(root, 1, L, sum);
    return sum;
}
```

2. 咕咕咕

17-18第一学期

一

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

7. B
8. A
9. A
10. B
11. C
12. D
13. D
14. A

二

1. head->next=null
2. 归并排序
3. 13
4. Kruskal
5. O(n)
6. 6;[2][3][2]
7. 不知道
8. 环
9. 之后
10. 4
11. n

三

1. 1)TR 2)993个 3)Tx是完全二叉树，从上往下，从左到右表示为：251，503，504，1007，1008，1009，1010，2015，2016

2.

0	1	2	3	4	5	6	7	8	9	10
11	22	46	13	01	70			41	31	30
成功的ASL：15/8										
失败的ASL：56/11										

3. (1)排出之前：95,85,49,72,34,40,43,58,65,20 排出并调整后：85,72,49,65,34,40,43,58,20,95 (2) 85,72,43,34,95,58,49,40,65,20 (3) 72,95,65,49,85,43,34,58,40,20

4. k=3i+j-n (n-1<=i+j<=n且i!=0时) k=0(i=0,j=n-1时)

5.

序号	S		D		P
0	1		0		
1	0		20		0,3,4,1
2	1		19		0,3,4,2
3	1		10		0,3
4	1		17		0,3,4
5	0		25		0,3,4,5

四

1.

```
//(1)
void createLinkedList(int* A, int n) {
    for (int i = n - 1; i >= 0; i++) {
        node* ptr = new node;
        ptr->val = A[i];
        ptr->next = pHead->next;
        pHead->next = ptr;
    }
}
//(2)
void reverse(node* pHead) {
    if (!pHead || !pHead->next)return;
    node* pre = nullptr, * cur = pHead->next, * next = cur->next;
    while (cur) {
        cur->next = pre;
        pre = cur;
        cur = next;
        next = cur->next;
    }
    pHead->next = pre;
}
```

2.
```
//(1)
bool isBST(node* p) {
    if (!p)return true;
    if ((p->left && p->val < p->left->val) || (p->right && p->val > p->right->val))return false;
    return isBST(p->left) && isBST(p->right);
}
//(2)
int totalLen(node* p, int len, int& sum) {//返回值是结点总数，len是当前结点的路径长度，sum是这棵树路径长度之和
    if (!p)return 0;
    sum += len;
    return 1 + totalLen(p->left, len + 1, sum) + totalLen(p->left, len + 1, sum);
}

double ASL(node* p) {
    int sum = 0;
    int nodeNum = totalLen(p, 1, sum);
    return sum / (double)nodeNum;
}
```

18-19第一学期

一

1. C
2. D
3. C
4. A
5. D
6. B
7. B
8. B
9. C
10. C
11. C
12. A
13. A

二

1. S==null
2. O(nd)
3. O(n^2^)
4. n+1;2
5. O(n^0.5^)
6. 9
7. 994
8. 73
9. 5
10. 3

三

1. (1)7 (2)4 (3)1388
2.

0	1	2	3	4	5	6	7	8	9	10
33	11	35	13	01	48			19	20	30
成功的ASL=17/8										

3. (1)126 (2)504,2018,1009,252,126,506,1013,253 (3)假设是完全二叉树，缺失的结点用#表示：504,252,1009,126,253,506,2018,#,#,#,#,#,#,1013
4. DFS时优先选择序号小的结点 DFS序列：V1,V2,V3,V7,V5,V6,V8,V4 生成树：(V1,V2),(V2,V3),(V3,V7),(V7,V5),(V7,V6),(V7,V8),(V2,V4)

	V1	V2	V3	V4	V5	V6	V7	V8
VE	0	6	14	9	29	25	18	20
VL	0	6	14	10	29	25	18	20

	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13
e(i)	0	0	6	6	9	9	14	14	6	18	20	18	18
l(i)	0	5	7	6	10	13	15	14	6	18	20	18	20
关键路径共四条，懒得打了													

5.

0	1	2	3	4	5	6	7	8	9	10	11
56	53	40	47	35	38	25	41	23	17	27	31

0	1	2	3	4	5	6	7	8	9	10	11
53	47	40	41	35	38	25	31	23	17	27	56

6. 什么鸟题，不写了

四

1.

```
void GetMinVal(node* pHead, int& minx) {
    if (!pHead)return;
    while (pHead->next) {
        pHead = pHead->next;
        if (pHead->val < minx)minx = pHead->val;
    }
}
//(2)
void SortLinkedList(node* pHead) {
    if (!pHead)return;
    node* pre = pHead, * tmp, * cur;//pre指向已排序部分的最后一个结点，cur指向被选择的结点的直接前驱
    while (pre->next) {
        tmp = pre->next;
        cur = pre;
        while (tmp->next) {//此处tmp用于遍历尚未排序的部分
            if (tmp->next->val > cur->next->val)cur = tmp;
            tmp = tmp->next;
        }
        tmp = cur->next;//此处tmp指向被选择的结点
        cur->next = cur->next->next;
        tmp->next = pre->next;
        pre->next = tmp;
        pre = pre->next;
    }
}
```

2. 和上一年的题目一样