

课程名称: 数据结构 C

一、选择题 (15 题, 每题 2 分, 共 30 分)

CCBBC DDAAB CCCCC

二、判断题 (对的打√, 错误打×, 10 题, 每题 1 分, 共 10 分)

×√√×√ ×√√×√

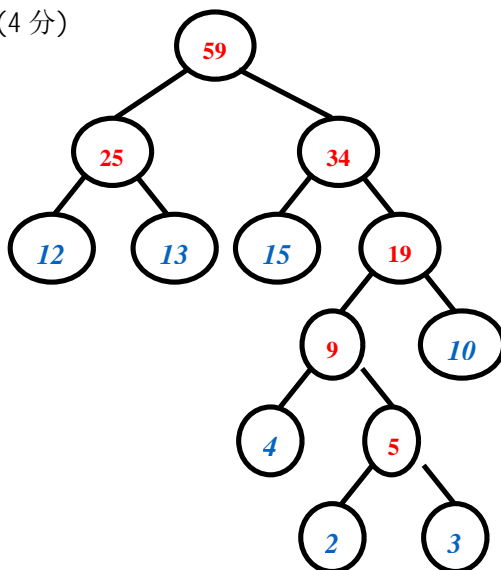
三、算法填空题 (填出横线处缺失的语句, 8 空, 每空 2 分, 共 16 分)

1. $p = p \rightarrow next$
2. $s \rightarrow next = p \rightarrow next$
3. $p \rightarrow next = s$
4. $(Q.rear+1) \% MAXQSIZE == Q.front$
5. $Q.rear = (Q.rear+1) \% MAXQSIZE$
6. !T 或 $T == NULL$
7. PreOrderEvenData(T->lchild)
8. PreOrderEvenData(T->rchild)

四、解答题 (3 题, 每题 8 分, 共 24 分)

1.

(1) (4 分)



(2) (2 分)

a: 111, b: 10, c: 00, d: 11011, e: 1100, f: 01, g: 11010

(3) (2 分)

编码长度 = $3 \times 10 + 2 \times 15 + 2 \times 12 + 5 \times 3 + 4 \times 4 + 2 \times 13 + 5 \times 2 = 151$

2.

(1) (4 分)

从 A 开始按普里姆算法加入最小生成树的边依次为:

AB、BE、EC、EH、HK、KG、AD、HF

(2) (2 分)

深度优先搜索遍历结果: ABECGKHFD

(3) (2 分)

广度优先搜索遍历结果: ABCDEFGHK

3

(1) (2 分)

$H(\text{on}) = 7, H(\text{hi}) = 6, H(\text{ok}) = 4, H(\text{if}) = 4,$

$H(\text{me}) = 7, H(\text{is}) = 6, H(\text{go}) = 0, H(\text{up}) = 4$

(2) (4 分)

序号	0	1	2	3	4	5	6	7	8	9	10
单词	go	up			ok	if	hi	on	me	is	at
查找长度	1	9			1	2	1	1	2	4	1

(3) (2 分)

$ASL = (1 \times 5 + 2 \times 2 + 4 \times 1 + 9 \times 1) / 9 = 22 / 9$

五、算法设计题(2题,第1题12分,第2题8分,共20分)

1.

(1) (6分)

```
void SortLinkedList(LinkedList L)
{
    LinkedList p, q, min;
    ElemType temp;

    for (p = L->next; p != NULL; p = p->next)
    {
        min = p;
        for (q = p; q != NULL; q = q->next)
            if (q->data < min->data)
                min = q;

        temp = min->data;
        min->data = p->data;
        p->data = temp;
    }
}
```

(2) (6分)

```
void RemoveDuplicate(LinkedList L)
{
    LinkedList p, q;

    for (p = L->next; p != NULL; )
    {
        if (p->next == NULL)
            break;
        if (p->data != p->next->data)
            p = p->next;
        else
        {
            q = p->next;
            p->next = q->next;
            free(q);
        }
    }
}
```

2.

```
void Remove(int C[], int j, int k, int &count)
{
    int i;

    //删除下标 j 的元素
    for (i = j + 1; i < count; i++)
        C[i - 1] = C[i];
    count--;
    //删除下标 k 的元素
    for (i = k + 1; i < count; i++)
        C[i - 1] = C[i];
    count--;
}
int JudgeCards(int C[27])
{
    int i, j, k;
    int card_in_hand[27], count;

    count = 0;
    for (i = 0; i < 27; i++)
    {
        //抓牌
        card_in_hand[count] = C[i];
        count++;

        //出掉对子
        for (j = 0; j < count; j++)
        {
            for (k = 0; k < count; k++)
                //如果有对子
                if (card_in_hand[j] == card_in_hand[k] && j != k)
                    Remove(card_in_hand, j, k, count);
        }
        //达到 8 张，返回当前抓过牌的数量 i
        if (count >= 8)
            return i + 1;
    }

    //执行到这里，说明没满 8 张，返回手上牌的数量
    return count;
}
```