

安徽大学 20 21 —20 22 学年第 2 学期

《 数据结构 》考试试卷（B 卷）答案

（闭卷 时间 120 分钟）

一、算法分析题（10 分）

1. 请分析如下算法的时间复杂度（每小题 2 分，共 6 分）

- (1) 答案: $O(n^2)$
(2) 答案: $O(\log_2 n)$
(3) 答案: $O(1)$

2. 将下面的算法补充完整：（每空 2 分，共 4 分）

答案: $s \rightarrow next = p \rightarrow next;$
 $p \rightarrow next \rightarrow prior = s;$

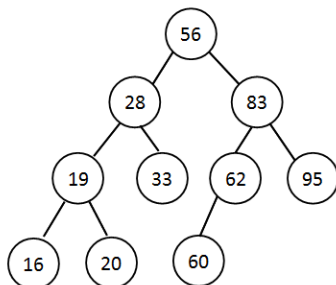
二、简答题（共 40 分）

1. 已知一个有向图的邻接表存储结构如下图所示，写出从顶点A出发的广度优先搜索序列（4分）

答案: AEDBFC

2. 已知10个元素{56, 28, 19, 83, 62, 95, 60, 16, 33, 20}，按照依次插入的方法生成一棵二叉排序树。（6分）

答案:



3. 答案:

((b, c, d), (d, e, (f, (g, h))))

(a, b)

head(Tail(head(Tail(Tail((head(Tail(Tail(L)))))))))))))

4. 答案: (1) (8分，每空0.5分，扣完为止)

哈希地址	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
关键字值	39	38	28	15	68	31	19				49		25	12	51
比较次数	1	5	1	2	2	1	1				1		1	2	3

(2) $ASL=20/13$ (2 分)

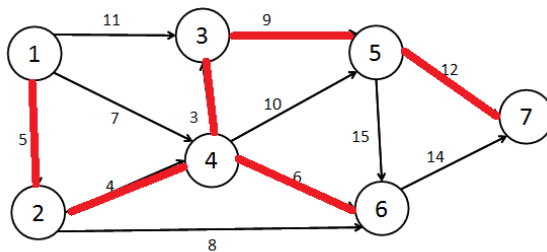
5. (11分) 用Prim算法求下面连通的带权图的最小代价生成树:

答案: (1)

$$\begin{bmatrix} 0 & 5 & 11 & 7 & \infty & \infty & \infty \\ \infty & 0 & \infty & 4 & \infty & 8 & \infty \\ \infty & \infty & 0 & \infty & 9 & \infty & \infty \\ \infty & \infty & 3 & 0 & 10 & 6 & \infty \\ \infty & \infty & \infty & \infty & 0 & 15 & 12 \\ \infty & \infty & \infty & \infty & \infty & 0 & 14 \\ \infty & \infty & \infty & \infty & \infty & \infty & 0 \end{bmatrix}$$

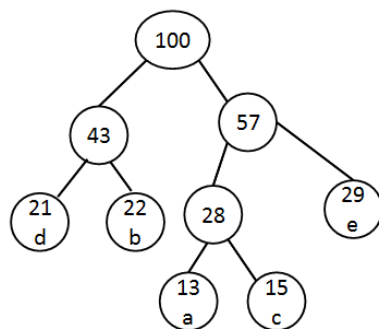
(2)

{u}	{v-u}	②	③	④	⑤	⑥	⑦
①	② ③ ④ ⑤ ⑥ ⑦	5	11	7	∞	∞	∞
①②	③ ④ ⑤ ⑥ ⑦		11	4	∞	8	∞
①②④	③ ⑤ ⑥ ⑦		3		10	6	∞
①②④③	⑤ ⑥ ⑦				9	6	∞
①②④③ ⑥	⑤ ⑦				9		14
①②④③ ⑥ ⑤	⑦						12
①②④③ ⑥ ⑤ ⑦							



三、应用题 (每小题 10 分, 共 20 分)

1、答案:



(1)

(2) a: 100

b: 01

c: 101

(5 分)

d: 00

e: 11

(3分)

(3) e d b a d c b (2分)

2、基因测序是一个更加复杂且耗时的过程，可以简单归纳为以下7个步骤：

答案：

next 的值为： 0 1 1 2 3

j	next[j]	步骤
j=1	0	初始情况
j=2	1	next[1]=0, k=0, next[2]=1
j=3	1	next[2]=1, k=0, next[3]=1
j=4	2	next[3]=1, k=1, next[4]=next[3]+1=2
j=5	3	next[4]=2, k=2, next[5]=next[4]+1=3

(5分)

nextval 的值为 01013 (5分)

四、算法设计题 (30分)

1、从一维数组 A[n] 中二分查找关键字为 K 的元素的递归算法，若查找成功则返回对应元素的下标，否则返回-1。

int Binsch(ElemType A[], int low, int high, KeyType K) (10分)

答案：

```
int Binsch(ElemType A[], int low, int high, KeyType K)
{
    if(low<=high) (1分)
    {
        int mid=(low+high) / 2; (2分)
        if(K==A[mid].key) return mid; (2分)
        else if (K<A[mid].key) return Binsch(A,low,mid-1,K); (2分)
        else return Binsch(A,mid+1,high,K); (2分)
    }
    else return -1; (1分)
} Binsch
```

2、二叉树采用二叉链表存储结构，类型定义如下：

```
typedef struct BiTNode{
    TelemType data; //结点数据域
    struct BiTNode *lchild,*rchild; //左右孩子指针
}BiTNode,*BiTree;
```

试设计算法：

(1) 按层次从上到下，每层从右到左的顺序列出二叉树所有结点的数据信息；(10分)

(2) 判定两棵二叉树是否相似。(10分)

答案:

```
(1) void lev_traverse(BiTNode *T)
{ BiTNode *q[100],*p;
  int head,tail;
  q[0]=T;head=0;tail=1;
  while(head<tail)
  { p=q[head++];
    printf("%c",p->data);
    if(p->rchild!=NULL)
    q[tail++]=p->rchild;
    if(p->lchild!=NULL)
    q[tail++]=p->lchild;
  }}

(2)
int like(BiTNode *t1, BiTNode *t2)
{
  int like1,like2;
  if(t1==t2&&t2==NULL) return(1);
  else
  if(t1==NULL &&t2!=NULL||t1!=NULL&&t2==NULL) return(0);
  else
  {
    like1=like(t1->lchild,t2->lchild);
    like2=like(t1->rchild,t2->rchild);
    return(like1&&like2);
  }
}
```