数据结构课程考试参考答案和评分标准

一、填空题(本大题共 15 小题 20 空，每空 1 分，共 20 分)

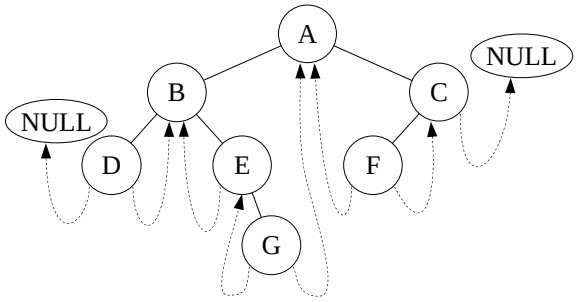
1. 数据元素
2. 线性、图状(网状)
3. 链式、散列
4. 基本操作集
5. 算法
6. O(n)
7. 栈、后进先出、队列、先进先出
8. 0x1118
9. $k=\frac{i(i-1)}{2}+j-1$
10. HFEADBGC
11. 11
12. 8
13. 6
14. 12
15. AEDCB

二、选择填空题(本大题共 15 小题，每小题 2 分，共 30 分)

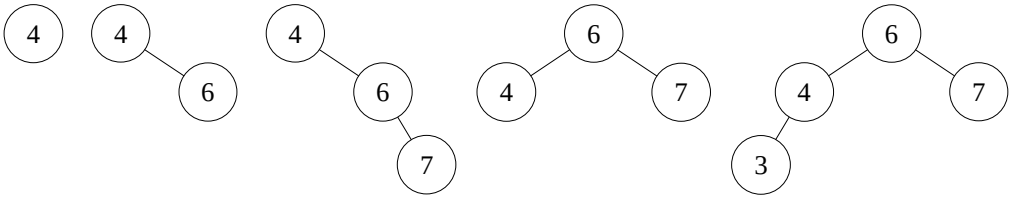
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	A	C	B	C	D	C	D	B	D	D	A	B	C	B

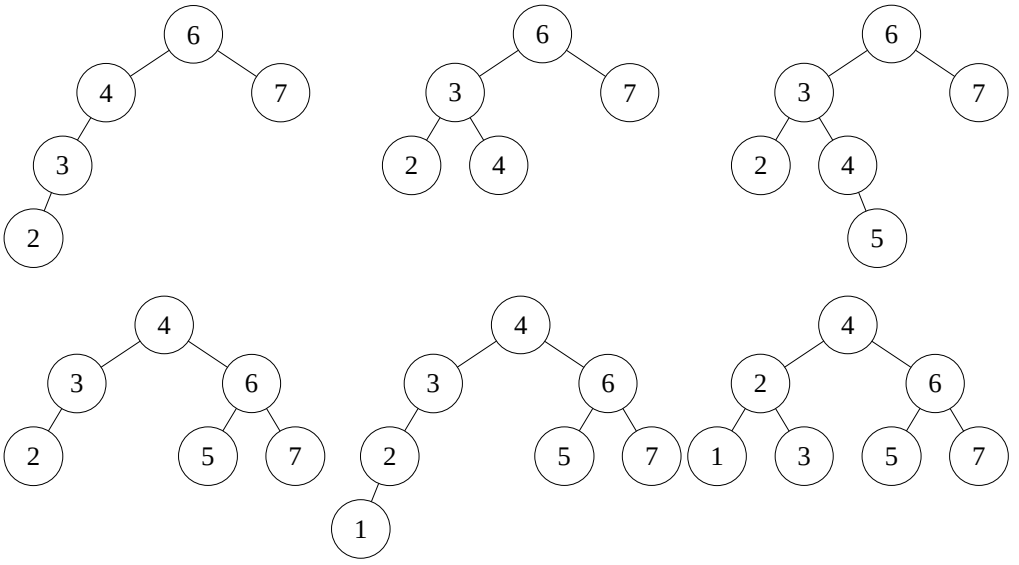
三、画图题(本大题共 3 小题，每小题 5 分，共 15 分)

1. 中根线索二叉树

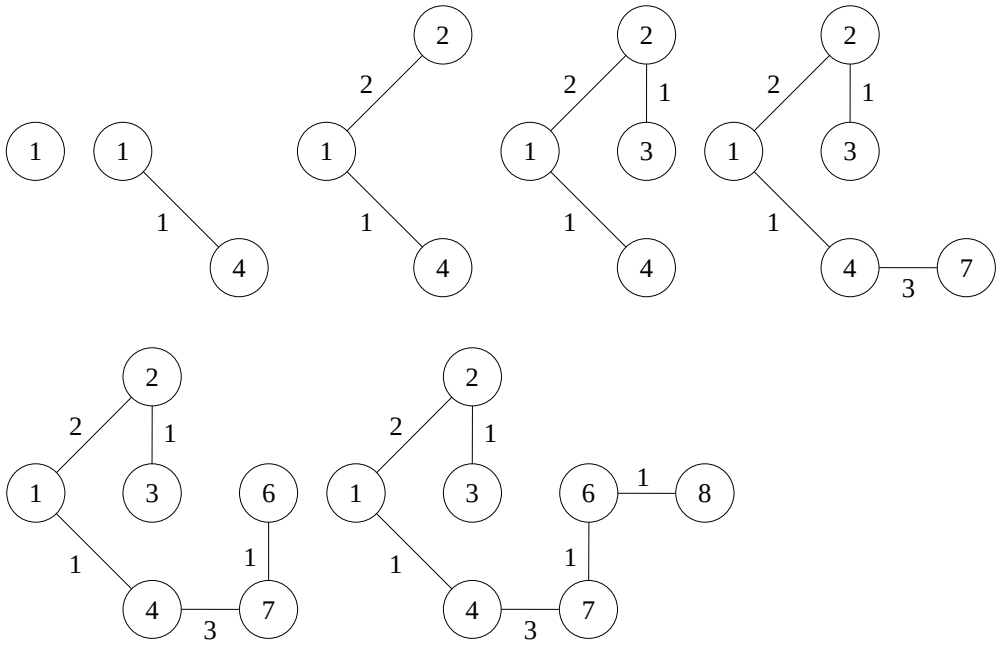


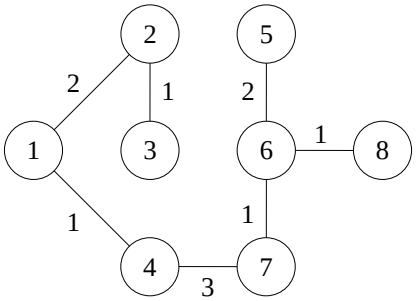
2. 平衡二叉树




6
4
7
3
2
6
3
7
2
4
6
3
7
2
4
5
4
3
6
2
5
7
4
3
6
2
5
7
1
4
2
6
1
3
5
7

3. 最小生成树(Prim)


1
1
1
4
2
2
1
1
4
2
2
1
1
3
1
4
2
2
1
1
3
1
4
7
3
2
2
1
1
3
6
1
4
7
3
2
2
1
1
3
6
1
8
1
4
7
3



四、分析题(本大题共 4 小题，每小题 5 分，共 20 分)

1. 简单选择排序

L = (2, 3, 5, 6, 7, 0, 1, 8, 9, 4)

L = (0, 3, 5, 6, 7, 2, 1, 8, 9, 4)

L = (0, 1, 5, 6, 7, 2, 3, 8, 9, 4)

L = (0, 1, 2, 6, 7, 5, 3, 8, 9, 4)

L = (0, 1, 2, 3, 7, 5, 6, 8, 9, 4)

L = (0, 1, 2, 3, 4, 5, 6, 8, 9, 7)

L = (0, 1, 2, 3, 4, 5, 6, 8, 9, 7)

L = (0, 1, 2, 3, 4, 5, 6, 8, 9, 7)

L = (0, 1, 2, 3, 4, 5, 6, 7, 9, 8)

L = (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)

2. 最短路径(Dijkstra)

步骤	顶点	距离					路径				
		A	B	C	D	E	A	B	C	D	E
0	A	0	1	∞	6	4	-	A	-	A	A
1	B	0	1	9	6	3	-	A	B	A	B
2	E	0	1	8	4	3	-	A	E	E	B
3	D	0	1	7	4	3	-	A	D	E	B
4	C	0	1	7	4	3	-	A	D	E	B

路径	长度	最短路径
A→B	1	A→B
A→C	7	A→B→E→D→C
A→D	4	A→B→E→D
A→E	3	A→B→E

3. 拓扑排序

   1, 2, 4, 7, 5, 3, 6, 8, 9

   1, 2, 4, 7, 5, 3, 8, 6, 9

   1, 2, 4, 7, 5, 8, 3, 6, 9

   1, 4, 2, 7, 5, 3, 6, 8, 9

   1, 4, 2, 7, 5, 3, 8, 6, 9

   1, 4, 2, 7, 5, 8, 3, 6, 9

4. 散列表

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
		15	2	3	5	31	18	16	4					

五、算法设计题(本大题共 2 小题，第 1 小题 9 分，第 2 小题 6 分，共 15 分)

1. 多项式求和(9 分)

   参考代码如下，时间复杂度为 O(m+n):

```
void AddPoly(struct POLY *h, const struct POLY *f, const struct POLY
*g)
{
   struct NODE *s, *p, *q, *r;
   double coef;
   int deg;

   // 首先将多项式 h 清为零多项式
   while ((s = h->head->next) != NULL)
   {
      h->head->next = s->next;
      delete s;
   }
   h->deg = -1;

   // 多项式相加
   p = f->head->next; q = g->head->next;
   r = h->head;
   while (p != NULL && q != NULL)
   {
      // 逐项计算系数、次数
      if (p->deg > q->deg)
      {
         coef = p->coef; deg = p->deg;
         p = p->next;
      }
      else if (p->deg < q->deg)
      {
         coef = q->coef; deg = q->deg;
         q = q->next;
      }
      else // p->deg == q->deg
      {
         coef = p->coef + q->coef; deg = p->deg;
         p = p->next; q = q->next;
      }
```

```
            // 插入结点
            if (coef != 0)
            {
               s = (struct NODE*)malloc(sizeof(struct NODE));
               s->coef = coef; s->deg = deg;
               r->next = s; r = s;
            }
         }

         // 添加剩余的项
         while (p != NULL)
         {
            s = (struct NODE*)malloc(sizeof(struct NODE));
            s->coef = p->coef; s->deg = p->deg;
            r->next = s; r = s;
            p = p->next;
         }

         while (q != NULL)
         {
            s = (struct NODE*)malloc(sizeof(struct NODE));
            s->coef = q->coef; s->deg = q->deg;
            r->next = s; r = s;
            q = q->next;
         }
         // 处理尾结点
         r->next = NULL;

         // 计算多项式次数
         if (h->head->next != NULL)
         {
            h->deg = h->head->deg;
         }
      }
```

考查点主要有: 1) 是否注意到要先清除原链表中的所有结点 2) 合并过程按次数由大到小进行, 指针移动是否正确 3) 当次数相同时求两系数的和, 且结果非零时才创建新结点来存储 4)是否注意到添加剩余的项 5) 是否对尾结点进行处理 6) 对多项式的次数进行调整。具体分值请阅卷教师酌情处理。

2. 求二叉树的深度(6 分)

参考代码如下, 时间复杂度为 O(*n*):

```
int Depth(NODE *root)
{
   int n = 0, d1, d2;
   if (root != NULL)
   {
      d1 = Depth(root->lch);
      d2 = Depth(root->rch);
      n = d1 >= d2 ? d1 : d2;
   }
   return n;
}
```

考查点主要有: 1) 是否考虑到空二叉树的情况 2) 是否正确调用递归函数 3) 是否重复调用降低了效率。具体分值请阅卷教师酌情处理。