

2016 年计算机学科专业基础综合试题参考答案

一、单项选择题

- | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|
| 1. B | 2. C | 3. A | 4. B | 5. B | 6. D | 7. B | 8. A |
| 9. B | 10. B | 11. D | 12. C | 13. C | 14. A | 15. D | 16. A |
| 17. C | 18. B | 19. A | 20. D | 21. D | 22. B | 23. D | 24. C |
| 25. B | 26. D | 27. B | 28. D | 29. B | 30. D | 31. B | 32. B |
| 33. A | 34. D | 35. B | 36. A | 37. D | 38. C | 39. A | 40. C |

二、综合应用题

41. 解析:

(1) 算法的基本设计思想

表达式树的中序序列加上必要的括号即为等价的中缀表达式。可以基于二叉树的中序遍历策略得到所需的表达式。
(3 分)

表达式树中分支结点所对应的子表达式的计算次序, 由该分支结点所处的位置决定。为得到正确的中缀表达式, 需要在生成遍历序列的同时, 在适当位置增加必要的括号。显然, 表达式的最外层 (对应根结点) 及操作数 (对应叶结点) 不需要添加括号。(2 分)

(2) 算法实现 (10 分)

将二叉树的中序遍历递归算法稍加改造即可得本题答案。除根结点和叶结点外, 遍历到其他结点时在遍历其左子树之前加上左括号, 在遍历完右子树后加上右括号。

```
void BtreeToE(BTree *root) {
```

```

    BtreeToExp(root, 1);                //根的高度为 1
}
void BtreeToExp( BTree *root, int deep)
{
    if(root == NULL) return;           //空结点返回
    else if(root->left==NULL&&root->right==NULL) //若为叶结点
        printf("%s", root->data); //输出操作数, 不加括号
    else{
        if(deep>1) printf("(");        //若有子表达式则加 1 层括号
        BtreeToExp(root->left, deep+1);
        printf("%s", root->data); //输出操作符
        BtreeToExp(root->right, deep+1);
        if(deep>1) printf(")");        //若有子表达式则加 1 层括号
    }
}
}

```

【评分说明】①若考生设计的算法满足题目的功能要求, 则 (1)、(2) 根据所实现算法的策略及输出结果给分, 细则见下表。

分数	备注
15	采用中序遍历算法且正确, 括号嵌套正确, 层数适当。
14	采用中序遍历算法且正确, 括号嵌套正确, 但括号嵌套层数 过多。例如, 表达式最外层加上括号, 或操作数加括号如 (a)。
11	采用中序遍历算法, 但括号嵌套层数不完全正确。例如, 左 右括号数量不匹配。
9	采用中序遍历算法, 但没有考虑括号。
≤7	其他

②若考生采用其他方法得到正确结果, 可参照①的评分标准给分。

③如果程序中使用了求结点深度等辅助函数, 但没有给出相应的实现过程, 只要考生进行了必要的说明, 可不扣分。

④若在算法的基本设计思想描述中因文字表达没有清晰反映出算法思路, 但在算法实现中能够表达出算法思想且正确的, 可参照①的标准给分。

⑤若算法的基本设计思想描述或算法实现中部分正确, 可参照①中各种情况的相应给分标准酌情给分。

⑥参考答案中只给出了使用 C 语言的版本, 使用 C++语言的答案参照以上评分标准。

42. 解析:

(1)Prim 算法属于贪心策略。算法从一个任意的顶点开始, 一直长大到覆盖图中所有顶点为止。算法每一步在连接树集合 S 中顶点和其他顶点的边中, 选择一条使得树的总权重增加最小的边加入集合 S。当算法终止时, S 就是最小生成树。

①S 中顶点为 A, 候选边为(A,D)、(A,B)、(A,E), 选择(A,D)加入 S。

②S 中顶点为 A、D, 候选边为(A,B)、(A,E)、(D,E)、(C,D), 选择(D,E), 加入 S。

③S 中顶点为 A、D、E, 候选边为(A,B)、(C,D)、(C,E), 选择(C,E)加入 S。

④S 中顶点为 A、D、E、C, 候选边为(A,B)、(B,C), 选择(B,C)加入 S。

⑤S 就是最小生成树。

依次选出的边为:

(A, D),(D, E),(C, E),(B, C) (4 分)

【评分说明】每正确选对一条边且次序正确, 给 1 分。若考生选择的边正确, 但次序不完全正确, 酌情给分。

(2)图 G 的 MST 是唯一的。(2 分) 第一小题的最小生成树包括了图中权值最小的四条边, 其他边都比这四条边大, 所以此图 MST 唯一。

(3)当带权连通图的任意一个环中所包含的边的权值均不相同, 其 MST 是唯一的。(2 分) 此题不要求回答充分必要条件, 所以回答一个限制边权值的充分条件即可。

②若考生给出的充分条件对图的顶点数和边数做了某些限制,例如,限制了图中顶点的个数(顶点个数少于3个)、限制了图的形状(图中没有环)等,则最高给1分。

43. 解析:

若 i 和 n 改为 int 类型，则不会出现死循环。（1 分）

(2) f1(23)与f2(23)的返回值相等。(1分) $f(23)=2^{23+1}-1=2^{24}-1$ ，它的二进制形式是24个1。int占32位，没有。float有1个符号位，8个指数位，23个底数位，23个底数位可以表示24位的底数。所以两者返回值相等。

f2(23)的机器数是 4B7F FFFFH。 (1 分)

(3) 当 $n=24$ 时, $f(24)=1\ 1111\ 1111\ 1111\ 1111\ 1111\ B$, 而 `float` 型数只有 24 位有效位, 舍入后数值增大, $f2(24)$ 比 $f1(24)$ 大 1。 (1 分)

因为 int 型最大可表示数是 0 后面加 31 个 1，故使 f1(n) 的返回值与 f(n) 相等的最大 n 值是 30。（1 分）

当 $n=126$ 时, $f(126)=2^{127}-1=1.1\cdots 1\times 2^{126}$, 对应阶码为 $127+126=253$, 尾数部分舍入后阶码加 1, 最终阶码为 254, 即 IEEE 754 单精度格式表示的最大阶码。故使 $f2$ 结果不溢出的最大 n 值为 126。(1 分)

【评分说明】对于第二问，只要给出 $n=23$ ，即可给分。对于第三问，只要给出 $n=126$ ，即可给分。

44. 解析:

M 的指令长短不一，不符合 RISC 指令系统特点。（1 分）

(3) $CF=1$ 。(1 分)

(4) f2 中不能用 shl 指令实现 power*2。 (1 分)

45. 解析:

(1) 函数 f1 的代码段中所有指令的虚拟地址的高 20 位相同，因此 f1 的机器指令代码在同一页中，仅占用 1 页。
(1 分) 页目录号用于寻找页目录的表项，该表项包含页表的位置。页表索引用于寻找页表的表项，该表项包含页的位置。

(2) push ebp 指令的虚拟地址的最高 10 位（页目录号）为 00 0000 0001，中间 10 位（页表索引）为 00 0000 0001，所以，取该指令时访问了页目录的第 1 个表项，（1 分）在对应的页表中访问了第 1 个表项。（1 分）

(3) 在执行 scanf() 的过程中，进程 P 因等待输入而从执行态变为阻塞态。（1 分）输入结束时，P 被中断处理程序唤醒，变为就绪态。（1 分）P 被调度程序调度，变为运行态。（1 分）CPU 状态会从用户态变为内核态。（1 分）

46. 解析：

先找出线程对在各个变量上的互斥、并发关系。如果是一读一写或两个都是写，那么这就是互斥关系。每一个互斥关系都需要一个信号量进行调节。

```
semaphore mutex_y1=1; //mutex_y1 用于 thread1 与 thread3 对变量 y 的互斥访问。
semaphore mutex_y2=1; //mutex_y2 用于 thread2 与 thread3 对变量 y 的互斥访问。
semaphore mutex_z=1; //mutex_z 用于变量 z 的互斥访问。
```

互斥代码如下：（5 分）

thread1	thread2	thread3
<pre> { cnum w; wait(mutex_y1); w = add(x, y); signal(mutex_y1); }</pre>	<pre> { cnum w; wait(mutex_y2); wait(mutex_z); w = add(y, z); signal(mutex_z); signal(mutex_y2); }</pre>	<pre> { cnum w; w.a = 1; w.b = 1; wait(mutex_z); z = add(z, w); signal(mutex_z); wait(mutex_y1); wait(mutex_y2); y = add(y, w); signal(mutex_y1); signal(mutex_y2); }</pre>

【评分说明】

①各线程与变量之间的互斥、并发情况及相应评分见下表。

线程对 变量	thread1 和 thread2	thread2 和 thread3	thread1 和 thread3	给分
x	不共享	不共享	不共享	1 分
y	同时读	读写互斥	读写互斥	3 分
z	不共享	读写互斥	不共享	1 分

②若考生仅使用一个互斥信号量，互斥代码部分的得分最多给 2 分。

③答案部分正确，酌情给分。

47. 解析：

(1) t_0 时刻到 t_1 时刻期间，甲方可以断定乙方已正确接收了 3 个数据帧，（1 分）分别是 S0,0、S1,0、S2,0。（1 分）R3,3 说明乙发送的数据帧确认号是 3，即希望甲发送序号 3 的数据帧，说明乙已经接收了序号为 0-2 的数据帧。

(2) 从 t_1 时刻起，甲方最多还可以发送 5 个数据帧，（1 分）其中第一个帧是 S5,2，（1 分）最后一个数据帧是 S1,2。（1 分）发送序号 3 位，有 8 个序号。在 GBN 协议中，序号个数 = 发送窗口 + 1，所以这里发送窗口最大为 7。此时已发送了 S3,0 和 S4,1，所以最多还可以发送 5 个帧。

(3) 甲方需要重发 3 个数据帧，（1 分）重发的第一个帧是 S2, 3。（1 分）在 GBN 协议中，接收方发送了 N 帧后，检测出错，则需要发送出错帧及其之后的帧。S2, 0 超时，所以重发的第一帧是 S2。已收到乙的 R2 帧，所以确认号应为 3。

(4) 甲方可以达到的最大信道利用率是：

$$\frac{7 \times \frac{8 \times 1000}{100 \times 10^6}}{0.96 \times 10^{-3} + 2 \times \frac{8 \times 1000}{100 \times 10^6}} \times 100\% = 50\% \quad (2 \text{ 分})$$

$U = \text{发送数据的时间} / \text{从开始发送第一帧到收到第一个确认帧的时间} = N * T_d / (T_d + RTT + T_a)$

U 是信道利用率，N 是发送窗口的最大值，T_d 是发送一数据帧的时间，RTT 是往返时间，T_a 是发送一确认帧的时间。这里采用捎带确认，T_d=T_a。

【评分说明】答案部分正确，酌情给分。