北 京 交 通 大 学 考 试 试 题（B 卷）

参 考 答 案 与 评 分 标 准

课程名称：算法设计与分析　　学年学期：2022—2023 学年第 2 学期

课程编号：M210004B　开课学院：软件学院　出题教师：刘铎，李令昆

第一部分、单项选择题。（共 16 分）

1.	2.	3.	4.	5.	6.	7.	8.
A	**C**	**C**	**B**	**B**	**C**	**D**	**B**

第二部分、计算题。（共 34 分）

9. （共 12 分）解：

变量说明（4 分）：用 x, y, z 分别表示小明对项目 A、B、C 的投资额（单位：万元）。则可得到线性规划模型如下（目标 3 分、等式组 3 分、变量类型约束 2 分）。

maximize $0.10x + 0.07y + 0.03z$

subject to

$x + y + z = 100$

$x \le (1/4)y$

$z \ge 0.35(x + y)$

$x \ge 0, y \ge 0, z \ge 0$

10. （共 10 分）解：

令 m=ln$n$，函数 $G(x)$=$T(\mathrm{e}^x)$。则$T(n) = G(m)$，$T(\sqrt{n}) = G(m/2)$。

有

$$G(m) = \begin{cases} 2G(m/2) + m & m > \ln c \\ O(1) & m \le \ln c \end{cases}$$

使用主定理得到 $G(m)\in\Theta(m\ln m)$，即 $F(n)\in\Theta(\ln n\cdot\ln\ln n)$。

11. （共 12 分）解：

（1）（6 分）令 $S[1]S[2]S[3]...S[n]$表示输入序列，$L(i)$（$1 \le i \le n$）表示以 $S[i]$结束的最长单调递减子序列的长度，$P(i)$表示这个最长单调递减子序列中在 $S[i]$之前一项（即倒数第二项）的位置。如果 $L(i) = 1$（即该子序列只包含 $S[i]$，而没有倒数第二项），则取 $P(i) = 0$。

（2）（2+2 分）过程如表 1 所示。

（3）（2 分）最长单调递减子序列是 10, 9, 8, 7, 6, 5。

表 1

i	1	2	3	4	5	6	7	8	9	10	11	12
$S[i]$	10	9	15	4	8	7	17	6	5	14	8	12
$L(i)$	1	2	1	3	3	4	1	5	6	2	3	3
$P(i)$	0	1	0	2	2	5	0	6	8	3	2	10

第三部分、综合分析题。（共 50 分）

12. （共 12 分）解：

算法思路：从 1 开始对堆中顶点进行标号，根的标号为 1，每个内部顶点（设其标号为 k）的左孩子编号为 $2k$，而其右孩子的编号 $2k+1$。

之后采用二分查找 1~n 中满足不是任何顶点的标号的最小值 m，m–1 就是堆中顶点的个数。此过程时间复杂度为 $O(\lg n)$。

当在堆中寻找标号为 x 的顶点时，根据 x 的二进制表示来判断向左孩子方向前进还是向右孩子方向前进。例如 11 的二进制表示为 1011，从第 2 位开始是 011，则从根开始依次访问当前顶点的左孩子、右孩子、右孩子即可找到标号为 11 的顶点。如果过程中遇到指针指向空（NULL），则表明标号为 x 的顶点在堆中不存在。

对于不超过 n 的 x，求其二进制可以通过不断除以 2 和记录余数完成，此过程时间复杂度为 $O(\lg n)$。之后从堆顶出发寻找标号为 x 的顶点的时间复杂度不超过树的高度，即此过程时间复杂度为 $O(\lg n)$。

（回答体现出正确的算法思路者，此题至少可得 6 分）

（1）（8 分）参考伪代码如下所示。

```
QExist ( x, root )
    y←x, i←0, node←root
    while (y>1) do
        route[i]←y mod 2
        i←i+1
        y←⌊y/2⌋
    for j = i-1 downto 0
        if route[j] = 0 then node←node.lch
        else node←node.rch
        if node=NULL then
            return FALSE
    return TRUE
```

```
COUNT ( root )
    low←1, high←n
    while (low≤high) do
        mid←(low+high)/2
        if (QExist(mid, root)) then
            low←mid+1;
        else high←mid-1;
    return high
```

（2）（4 分）**COUNT** 执行时间复杂度为 $O(\ln n)$，设 **FindFirst**/**FindLast** 的时间复杂度为 $T(n)$（二者流程类似），则 $T(n)=T(n/2)+O(m)$，$T(1)\in O(m)$，即 $T(n)\in O(m\log n)$。因此 **COUNT-PREFIX** 的时间复杂度为 $O(m\log n)$。

13. （共 12 分）解：

定义：若在完成任务的顺序 $i_1, i_2, \cdots, i_n$ 中存在 $1\le s<t\le n$ 且 $b_{i_s}-a_{i_s}>b_{i_t}-a_{i_t}$，则称此时$(i_a, i_b)$构成一个逆序。

如果一个完成任务的顺序中有一对逆序，那么就一定存在两个相邻的任务形成逆序。（学生可直接使用此结论而不加证明。）

假设 S^*是保证完所有任务且达到初始能量最小值的所有解中逆序数最小的解之一，其启动任务顺序为 $i_1, i_2, \cdots, i_n$。

①若 S^*中不存在逆序，则其就是贪婪解。

②如果 S^*中存在逆序(i_s, i_{s+1})，即$b_{i_{s+1}}-a_{i_{s+1}}>b_{i_s}-a_{i_s}$。

假设在时刻 x 时启动了任务 i_s，则有$P\ge b_{i_s}$且$P-a_{i_s}\ge b_{i_{s+1}}$。

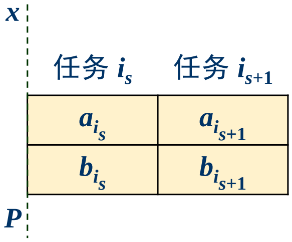
则$P\ge a_{i_s}+b_{i_{s+1}}>b_{i_{s+1}}$，即在时刻 x 时也可以启动任务 i_{s+1}。

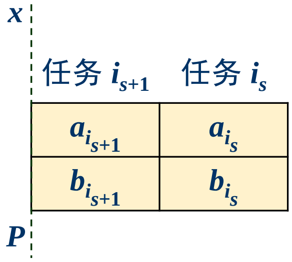
假设在时刻 x 时启动任务 i_{s+1}，执行完成任务 i_{s+1} 后，剩余能量为$P-a_{i_{s+1}}>P+b_{i_s}-a_{i_s}-b_{i_{s+1}}\ge\left(P-a_{i_s}-b_{i_{s+1}}\right)+b_{i_s}\ge b_{i_s}$，即也可以启动任务 i_s。

而在初始能量数相同的情况下，也可以先启动任务 i_{s+1} 再启动任务 i_s。此时总逆序数严格减少。

而交换任务 i_s 和任务 i_{s+1} 的执行次序导致总逆序数严格减少。

于是和 S^*的假设产生矛盾。





14. （共 13 分）**解：**

（此题若计算 LCS 或 SCS，则不得分。）

（1）（3+5 分）令 $X[1]X[2]X[3]...X[n]$表示输入序列 X，$Y[1]Y[2]Y[3]...Y[m]$表示输入序列 Y，$\mathrm{ED}(i, j)$表示序列 $X[1]X[2]...X[i]$表示输入 X，$Y[1]Y[2]...Y[j]$的编辑距离。问题目标为 $\mathrm{ED}(m, n)$。

递推式和初值如下所示：

$$\mathrm{ED}(i,j)=\begin{cases} 5j & ,\text{if } i=0 \\ \min\begin{cases} c_{x_i y_j}+\mathrm{ED}(i-1,j-1) \\ 5+\mathrm{ED}(i-1,j) \\ 5+\mathrm{ED}(i,j-1) \end{cases} & ,\text{otherwise} \\ 5i & ,\text{if } j=0 \end{cases}$$

其中 $c_{x_i y_j}$ 表示字符 x_i 和字符 y_j 错误匹配的代价（后文伪代码中的 cost 变量）。

伪代码为：

```
Edit-Distance ( X, Y, m, n )
    for i = 0 to m
        ED(i, 0)  ← 5×i
    for j = 0 to n
        ED(0, j)  ← 5×j
    for i = 1 to m
        for j = 1 to n
            ED(i, j) ← min{ cost(X[i], y[j]) + ED[i-1, j-1],
                             5 + ED[i-1, j], 5 + ED[i, j-1] }
    return ED(m, n)
```

（2）（3+2 分）过程如表 1 所示，二者的编辑距离为 10。

表 2

X \ Y	“”	A	C	D	B
“”	0	5	10	15	20
C	5	1	5	10	15
B	10	6	3	8	10
A	15	10	7	7	11
D	20	15	12	7	10

15. （共 13 分）**解：**

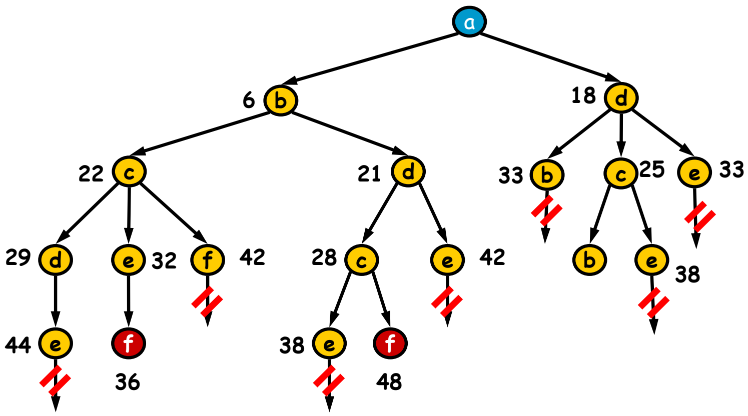
（1）（5 分）用 $C_a(v)$表示从顶点 a 到当前顶点 v 的道路长度，C_e 表示图中最短边的长度（即为 4），则估界函数为 $C_a(v)+C_e$。

维护“到目前为止，从 a 到 f 的道路长度的最小值” *current_best*。

若 $v=f$ 且 $C_a(v)<current\_best$，则更新 *current_best* 为 $C_a(v)$。

若 $i\neq n$，且有 $C_a(i)+C_e\geq current\_best$，则进行剪枝。

（2）（6 分）以一条显然的道路（“最上路线”）的长度 42 为初始界。剪枝后的（部分）搜索树如图 2 所示。



（3）（2 分）从 a 到 f 的最短道路为 a, b, c, e, f，总长度为 36。