

17-18 学年《算法设计与分析》期中参考答案和评分标准

一、单选题（每题 2 分，共 20 分）

BCAAC CBABA

二、填空题（每题 2 分，共 20 分）

1. P 类 (或者 易解)
2. 旧值推出新值
3. 局部
4. 不独立的(或者 重叠的)
5. 蛮力法
6. 贪心法
7. 深度优先
8. 剪枝
9. 堆
10. 拉斯维加斯

三、算法应用题（每题 10 分，共 40 分）

1. (1) 设 $n = 4^k$, 代入 $T(n)$ 展开右式有:

(共 8 分)

$$\begin{aligned}
 T(n) &= 7T(4^{k-1}) + 3 \cdot 4^k = 7[7T(4^{k-2}) + 3 \cdot 4^{k-1}] + 3 \cdot 4^k \\
 &= 7^2 T(4^{k-2}) + 3 \cdot 7 \cdot 4^{k-1} + 3 \cdot 4^k \\
 &= 7^3 T(4^{k-3}) + 3 \cdot 7^2 \cdot 4^{k-2} + 3 \cdot 7 \cdot 4^{k-1} + 3 \cdot 4^k \\
 &= \dots \\
 &= 7^k T(1) + 3[4^k + 4^{k-1} \cdot 7 + 4^{k-2} \cdot 7^2 + \dots + 4^1 \cdot 7^{k-1}] \\
 &= 7^k + 4(7^k - 4^k)
 \end{aligned}$$

因为 $n = 4^k$, 所以 $7^k = n^{\log_4 7}$

最终得: $T(n) = 5n^{\log_4 7} - 4n$ 。

- (2) 时间复杂度 $O(n^{\log_4 7})$

(2 分)

2. 将这 10 位客户的申请按照结束时间 $f(i)$ 递增排序, 如下表:

i	9	1	6	2	10	4	7	3	5	8
$s(i)$	0	2	4	3	8	6	9	8	11	12
$f(i)$	3	4	5	7	9	10	10	11	12	15

- 1) 首先选择申请 9(0, 3) (3 分)
- 2) 依次检查后续客户申请, 只要与已选择的申请相容不冲突, 则选择该申请, 直到所有申请检查完毕:
可以依次选择申请 6(4, 5)、申请 10(8, 9)、申请 7(9, 10)、5(11, 12)、8(12, 15)。(3 分)
- (2) 最大相容集为(9, 6, 10, 7, 5, 8) 或者(9, 2, 10, 7, 5, 8), 可以满足 6 位客户需要 (2 分)
3. 一种产品一个阶段 (每个阶段过程各 2 分, 最佳方案 2 分)。

第 1 阶段: 只生产 A 产品

设备	0	1	2	3	4	5	6
A	0	4	5	7	7	8	9
方案	0	1	2	3	4	5	6

第 2 阶段: 同时生产 A 和 B 产品

设备	0	1	2	3	4	5	6
AB	0	4	6	8	10	13	14
方案	00	10	11	12	13	14	24

第 3 阶段: 同时生产 A、B 和 C 产品

设备	0	1	2	3	4	5	6
ABC	0	4	7	9	11	14	16
方案	000	100	101	111	103 121	104	114 141

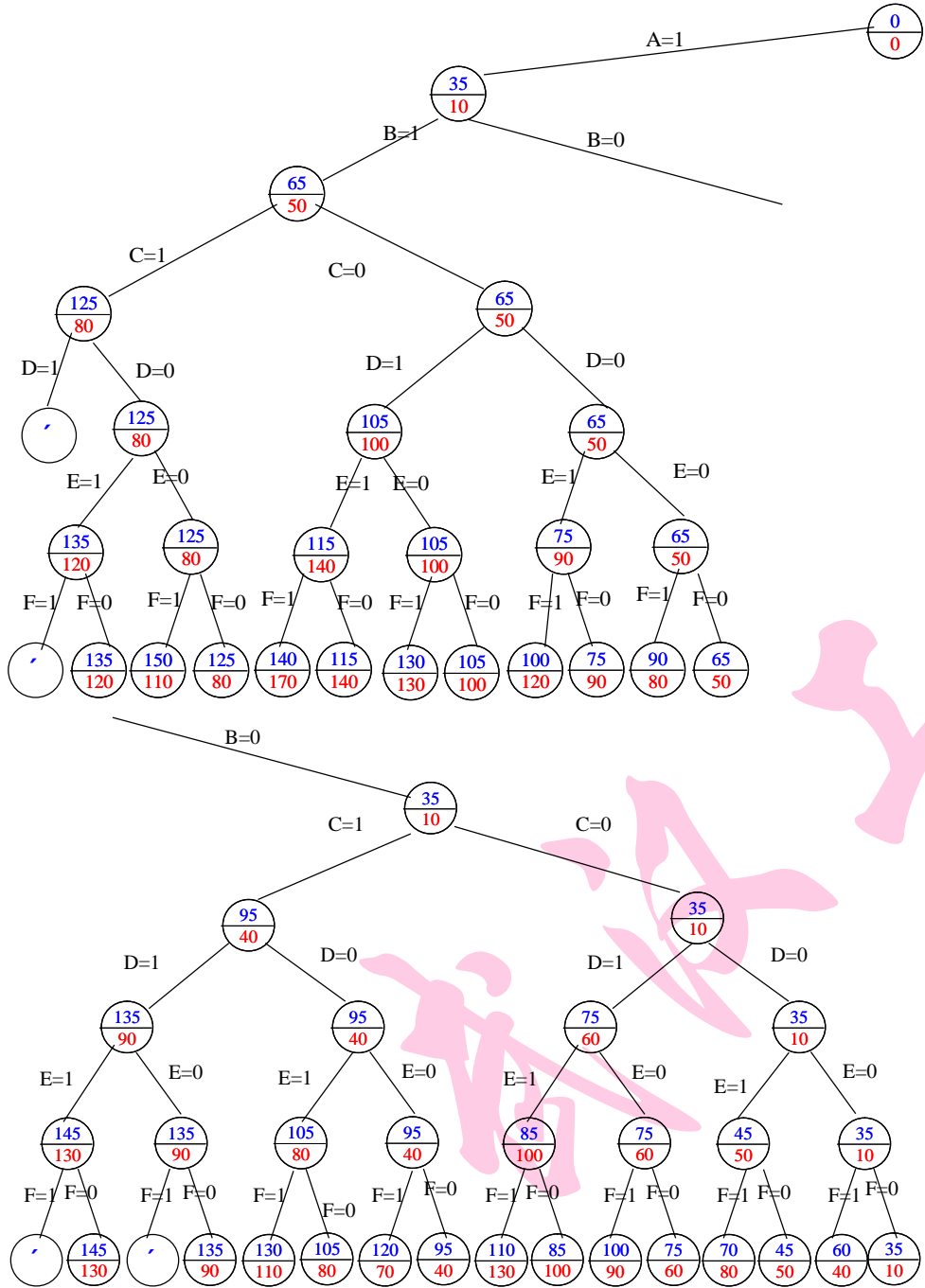
第 4 阶段: 同时生产 A、B、C 和 D 产品

设备	0	1	2	3	4	5	6
ABCD	0	4	8	11	13	15	18
方案	0000	1000 0001	1001	1011	1111	1031 1211	1041

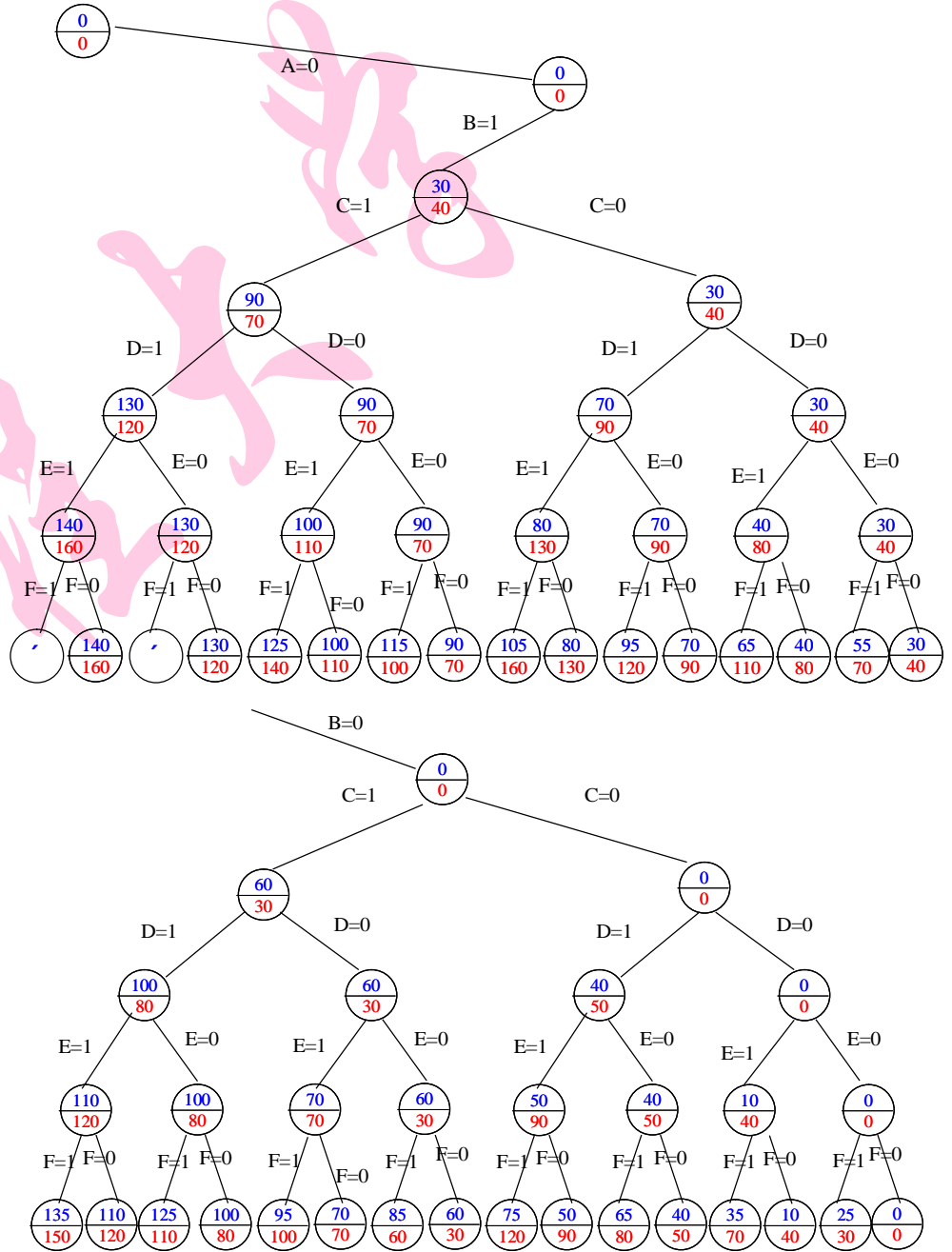
最大利润为 18 万元。

分配方案: A 产品 1 台、B 产品 0 台、C 产品 4 台、D 产品 1 台。

4. 子集树比较复杂，以下是左边半枝树：



以下是右边半枝树：



从树中可知最大价值 170，方案为选择 A、B、D、E、F。

四、算法设计题（使用 C 或 C++ 或 Java 语言实现）（每题 10 分，共 20 分）

1. 使用 3 等分减治法完成，时间复杂度 $O(\log_3 N)$ ，伪码如下：

1° 将 N 块金币等分为 3 部分 a, b, c

设 $t = \frac{N}{3}$

如果 $N \equiv 0 \pmod{3}$ ，则 $a \leftarrow t, b \leftarrow t, c \leftarrow t$

否则 如果 $N \equiv 1 \pmod{3}$ ，则 $a \leftarrow t, b \leftarrow t, c \leftarrow t + 1$

否则 $a \leftarrow t + 1, b \leftarrow t + 1, c \leftarrow t$

2° 比较 a, b 的重量

如果 $a = b$ ，则 $N \leftarrow c$

否则 如果 $a < b$ 则 $N \leftarrow a$

否则 $N \leftarrow b$

3° 如果 $N > 1$ ，转 1° 继续执行

否则，已经求得答案，退出。

算法补充说明： 问题的前提已经保证，如果最后只剩下 2 块金币，则

2° 中不会出现 $a = b$ 。

2. 考虑到问题特征，使用递归较容易实现，以下是完整的 C++ 程序。

```
#include <algorithm>
```

```
#include <iostream>
```

```
using namespace std;
```

```
void divide(int a[], int last, int m, int n, int subscript, int &cnt)
```

```
{
```

```
    if (m == 0)        // 放到 0 了，计数加 1
```

```
        cnt ++;
```

```
    else if (subscript < n) // 放下的数字个数不够
```

```
    {
```

```
        for (int i = min(last, m); i >= 1; i --)
```

```
        { // 待放下数不能超过左边紧挨着的数
```

```
            a[loc] = i;
```

```
            divide(a, i, m - i, n, subscript + 1, cnt);
```

```
        }
```

```
    }
```

```
}
```

```
int main()
```

```
{
```

```
    int data[100] = {0}, count = 0, m, n;
```

```
    cin >> m >> n;
```

```
    divide(data, m, m, n, 0, count);
```

```
    cout << count << endl;
```

```
    return 0;
```

```
}
```