算法基础

第二次作业参考答案（DDL：2024 年 9 月 25 日 23:59）

Q1.($10 + 10 = 20$ 分)

1. 对于递归式 $T(n) = 4T(n/2) + n^2 \log^2 n$，给出 $T(n)$ 的一个渐近紧确界;
2. 对于递归式 $T(n) = T(2n/3) + 2^{\lceil \log n \rceil}$，给出 $T(n)$ 的一个渐近紧确界。

Solution:

1. **主定理**：令 $a \geq 1$ 和 $b > 1$ 是常数，$f(n)$ 是一个函数，$T(n)$ 是定义在非负整数上的递归式：
$$T(n) = aT(n/b) + f(n)$$
若对整数 $k \geq 0$ 有 $f(n) = \Theta(n^{\log_b a} \log^k n)$,则 $T(n) = \Theta(n^{\log_b a} \log^{k+1} n)$。
应用主定理, $a = 4$,$b = 2$,$n^{\log_b a} = n^2, k = 2$,
解为 $\Theta(n^2 \log^3 n)$。
2. 定义 $F(n) = F(2n/3) + n, G(n) = G(2n/3) + 2n$，则由主定理显然有：$F(n) = \Theta(n), G(n) = \Theta(n)$，而 $n = 2^{\log n} \leq 2^{\lceil \log n \rceil} \leq 2^{(\log n)+1} = 2n$，因此 $F(n) \leq T(n) \leq G(n)$，故 $T(n) = \Theta(n)$。

Q2. ($5 + 5 + 10 + 10 + (10) = 30 + (10)$ 分)

问题背景：假设我们有一个大型音视频文件，需要对其进行编码处理。为了提高编码效率，我们采用分治法将文件分割成较小的部分进行处理，然后合并这些编码后的部分。具体策略如下：

将文件分为 **2** 个大小为原文件的 $\frac{1}{4}$ 的子文件和 **3** 个大小为原文件的 $\frac{1}{6}$ 的子文件。对这些子文件分别进行递归编码处理。合并这些处理后的子文件，合并过程所需时间为 $n \log n$; 对于大小不超过 1 的文件需要花费 1 单位的时间处理。请回答以下问题：

1. 写出音视频文件的编码处理时间的递推公式 $T(n)$; 假设文件的初始大小为 $n = 24$，计算编码处理所需的总时间 $T(24)$。
2. 试编写程序分别计算 $T(100)$，$T(1000)$，$T(10000)$, 若已知 $T(n)$ 的时间复杂度为 $\Theta(n^a \log^b n)$, 尝试猜测 $T(n)$ 的形式。

3. (使用替代法) 分析并给出编码处理时间 $T(n)$ 的渐近时间复杂度。

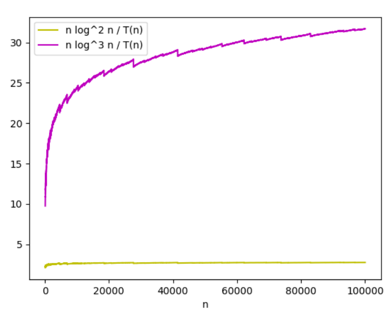
4. (使用递归树法) 分析并给出编码处理时间 $T(n)$ 的渐近时间复杂度。

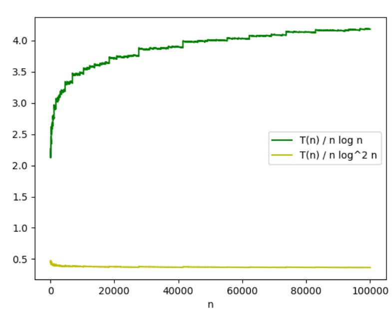
5. * (参考教材第 4 章注记, 使用 Akra-Bazzi 定理) 分析并给出编码处理时间 $T(n)$ 的渐近时间复杂度。

Solution:

1. $T(n) = 2T(\frac{n}{4}) + 3T(\frac{n}{6}) + n\log n$,$T(24) = 139$.

2. $T(100) = 1001$,$T(1000) = 19008$,$T(10000) = 320825$. 分别画出 $T(n)$ 与 $n\log n$、$n\log^2 n$、$n\log^3 n$ 在 n 从 100 到 10000 相比较的曲线如下:





可以发现 $\frac{n\log^2 n}{T(n)}$ 与 $\frac{T(n)}{n\log^2 n}$ 始终较为平稳，猜测 $T(n) = \Theta(n\log^2 n)$.

```
def encode(n):
    if n <= 1:
        return 1

    T1 = encode(n // 4)
    T2 = encode(n // 6)

    merge_cost = n * math.log2(n)
    return 2 * T1 + 3 * T2 + merge_cost
```

3. 猜测 $T(n) = \Theta(n\log^2 n)$.

取 $c > \frac{1}{2+\log 6}$, 假设假设 $T(k) \le ck\log^2 k$, 对 $k \le n-1$ 成立，则

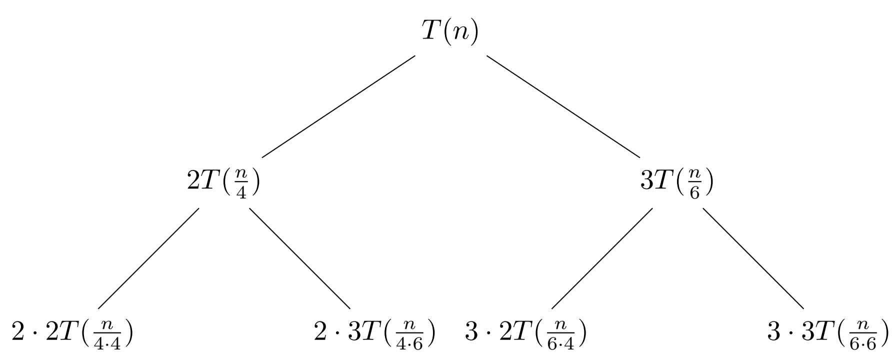
$$\begin{aligned}
T(n) &= 2T(\frac{n}{4}) + 3T(\frac{n}{6}) + n\log n \\
&\le 2c\frac{n}{4}\log^2\frac{n}{4} + 3c\frac{n}{6}\log^2\frac{n}{6} + n\log n \\
&= cn\log^2 n - (2c + \log 6 * c - 1)n\log n + (2c + \frac{\log^2 6}{2}c)n \\
&\le cn\log^2 n
\end{aligned}$$

取 $c < \frac{1}{2+\log 6}$, 假设 $T(k) \ge ck\log^2 k$, 对 $k \le n-1$ 成立，则

$$\begin{aligned}
T(n) &= 2T(\frac{n}{4}) + 3T(\frac{n}{6}) + n\log n \\
&\ge 2c\frac{n}{4}\log^2\frac{n}{4} + 3c\frac{n}{6}\log^2\frac{n}{6} + n\log n \\
&= cn\log^2 n - (2c + \log 6 * c - 1)n\log n + (2c + \frac{\log^2 6}{2}c)n \\
&\ge cn\log^2 n
\end{aligned}$$

综上，$T(n) = \Theta(n\log^2 n)$

4. 递归树如下



第 h 层的节点可以整理为

$$\binom{h}{a} \times 2^a \cdot 3^{h-a} \times T(\frac{n}{4^a \cdot 6^{h-a}}), a = 0, 1, 2, \cdots, h$$

第 h 层所需的时间开销为

$$\begin{aligned}
&\sum_{a=1}^{h}\binom{h}{a}\times 2^a\cdot 3^{h-a}\times\frac{n}{4^a\cdot 6^{h-a}}\log\frac{n}{4^a\cdot 6^{h-a}}\\
=&\sum_{a=1}^{h}\binom{h}{a}\times\frac{2^a\cdot 3^{h-a}}{4^a\cdot 6^{h-a}}\times n(\log n-\log 4^a\cdot 6^{h-a})\\
=&\sum_{a=1}^{h}\binom{h}{a}\times\frac{1}{2^h}\times n(\log n-\log 4^a\cdot 6^{h-a})\\
=&n\log n-\frac{1}{2^h}\sum_{a=1}^{h}\binom{h}{a}\times\log 4^a\cdot 6^{h-a}\times n
\end{aligned}$$

而

$$\begin{aligned}
&\frac{1}{2^h}\sum_{a=1}^{h}\binom{h}{a}\times\log 4^a\cdot 6^{h-a}\times n\\
\leq&\frac{1}{2^h}\sum_{a=1}^{h}\binom{h}{a}\times\log 6^a\cdot 6^{h-a}\times n\\
=&\log 6\cdot h\times n
\end{aligned}$$

且

$$\begin{aligned}
&\frac{1}{2^h}\sum_{a=1}^{h}\binom{h}{a}\times\log 4^a\cdot 6^{h-a}\times n\\
\geq&\frac{1}{2^h}\sum_{a=1}^{h}\binom{h}{a}\times\log 4^a\cdot 4^{h-a}\times n\\
=&2\cdot h\times n
\end{aligned}$$

故第 h 层所需的时间开销为 $\Theta(n\log n+h\cdot n)$,$h=0,1,\cdots,\log n$，总时间开销为 $\Theta(n\log^2 n)$

5. 由 Akra-Bazzi 定理,$p = 1$,

$$\begin{aligned}
T(x) &= \Theta(x(1 + \int_1^x \frac{u \log u}{u^2} du)) \\
&= \Theta(x(1 + \int_0^{\log x} v dv)), \qquad (\text{let} \quad v = \log u) \\
&= \Theta(x(1 + \int_0^{\log x} v dv)) \\
&= \Theta(x + x \frac{\log^2 x}{2}) \\
&= \Theta(x \log^2 x)
\end{aligned}$$

Q3. (15 分)

由于 MAX-HEAPIFY 的最后一行的递归调用可能会损失效率，请用循环控制取代递归，重写 MAX-HEAPIFY 代码为 NEW-MAX-HEAPIFY，并给出 MAX-HEAPIFY 和 NEW-MAX-HEAPIFY 的时间复杂度。

```
MAX-HEAPIFY(A, i)
    l = LEFT(i)
    r = RIGHT(i)
    if l <= A.heap-size and A[l] > A[i]
        largest = l
    else largest = i
    if r <= A.heap-size and A[r] > A[largest]
        largest = r
    if largest != i
        exchange A[i] with A[largest]
        MAX-HEAPIFY(A, largest)
```

Solution:

```
NEW-MAX-HEAPIFY(A, i)
    while true
        l = LEFT(i)
        r = RIGHT(i)
        if l <= A.heap-size and A[l] > A[i]
            largest = l
        else largest = i
        if r <= A.heap-size and A[r] > A[largest]
            largest = r
        if largest == i
            return
        exchange A[i] with A[largest]
        i = largest
```

MAX-HEAPIFY:

对于 MAX-HEAPIFY 来说，每次可以通过 $\Theta(1)$ 的时间将问题转化为左子树或右子树上的 MAX-HEAPIFY 问题。而左子树的规模永远比右子树

大，最大为原问题规模的 $\frac{2}{3}$, 故 $T(n) \leq T(2n/3) + \Theta(1)$。由主定理，解得 $T(n) = O(\log n)$

NEW-MAX-HEAPIFY:

在最坏情况下，需要遍历堆的高度。完全二叉树的高度为 $\log n$。

每次交换操作的时间复杂度是 $O(1)$，因为只是进行简单的比较和赋值操作。

因此，总的时间复杂度为：$O(\log n)$

Q4.（15 分）
试证明：在一个随机输入数组上，对于任何常数 $0 < \alpha \leq 1/2$，**Partition** 产生比 $1 - \alpha : \alpha$ 更平衡的划分的概率约为 $1 - 2\alpha$.
Solution: 假设数组长度为 n，在两个点可能产生 $1 - \alpha : \alpha$ 的划分：αn, $(1 - \alpha)n$；产生比 $(1 - \alpha) : \alpha$ 更差的划分的点是 $1 \cdots \alpha n$ 和 $(1 - \alpha)n \cdots n$，占所有点个数的比率为 2α，所以产生更对称划分的概率为 $1 - 2\alpha$。

Q5.（20 分）在 $[2,3,3,5,6,7,1]$ 数组上对于不同基准选取策略，应用快速排序。分别写出以最右边为基准、三数取中、最均衡划分的快速排序步骤。(三数取中选取最左边、最右边、最中间三个值的中位数作为基准; 最均衡划分选取使得划分最均衡的数作为基准)

Solution:

以最右边为基准

- [2,3,3,5,6,7,1]
- 1,[3,3,5,6,7,2]
- 1,2,[3,5,6,7,3]
- 1,2,3,3,[6,7,5]
- 1,2,3,3,5,[6,7]
- 1,2,3,3,5,6,7

三数取中

- [2,3,3,5,6,7,1]
- 1,2,[3,3,5,6,7]
- 1,2,3,3,[5,6,7]
- 1,2,3,3,5,6,7

最均衡划分

- [2,3,3,5,6,7,1]
- [2,3,1],3,[5,6,7]
- 1,2,3,4,5,6,7