期中样卷答案

1-1

Master Theorem: Let $a \geq 1$ and $b > 1$ be constants and $f(n)$ be a function. Let $T(n)$ be defined on the nonnegative integers by the following recurrence

$$T(n) = aT(n/b) + f(n)$$

Notice that here n/b can be interpreted as either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$. Then $T(n)$ can be bounded asymptotically as follows:

1. If there exists a constant $\varepsilon > 0$ such that $f(n) = O(n^{\log_b a - \varepsilon})$ then $T(n) = \Theta(n^{\log_b a})$.
2. If there exists an integer $k \geq 0$ such that $f(n) = \Theta(n^{\log_b a} \lg^k n)$ then $T(n) = \Theta(n^{\log_b a} \lg^{k+1} n)$.
3. If there exists a constant $\varepsilon > 0$ such that $f(n) = \Omega(n^{\log_b a + \varepsilon})$, and if $af(n/b) \leq cf(n)$ for some constant $c < 1$, then $T(n) = \Theta(f(n))$.

1. 递归式 $T(n) = 7T(\frac{n}{2}) + n^2$ 的解为 $T(n) = \Theta(n^2)$。 (　　)
The solution of the recurrence $T(n) = 7T(\frac{n}{2}) + n^2$ is $T(n) = \Theta(n^2)$.

False

$\log_b a = \log_2 7 > 2$，故存在 $\epsilon > 0$ 满足 $f(n) = n^2 = O(n^{\log_2 7 - \epsilon})$，$T(n) = \Theta(n^{\log_2 7})$

1-2

2. 给定一个包含 n 个整数的数组 A，归并排序总是可以在最坏情况下用 $O(n \lg n)$ 的时间对数组 A 进行排序。 (　　)
Given an array A of n integers, merge sort can always sort the array A in time $O(n \lg n)$ in the worst case.

True

1-3

3. 假设有一个包含 n 个待排序数据记录的数组，且每条记录的关键字的值为 0 或 1。对这样一组记录进行排序，存在时间代价为 $O(n)$，稳定的原址（除了输入数组外，算法只需要固定的额外存储空间）排序算法。 ( )
Suppose that we have an array of n data records to sort and that the key of each record has the value 0 or 1. There exists such an algorithm for sorting such a set of records that satisfies the following three characteristics: 1) The algorithm runs in $O(n)$ time. 2) The algorithm is stable. 3) The algorithm sorts in place, using no more than a constant amount of extra storage space.

True

1. 统计 0 和 1 出现次数，并记录下来，不妨假设 0 出现 a 次，那么 1 出现 $n-a$ 次。
2. 将数组遍历一遍，将所有的 0 元素从左到右依次替换为 $1, 2, 3, ..., a$；将所有的 1 元素替换为 $a+1, a+2, ..., n$。
3. 从左到右使用交换的方式，将每个元素放到合适的位置（合适指 $A[i]=i$）。
4. 将 $1, 2, 3, ..., a$ 替换为 0；$a+1, a+2, ..., n$ 替换为 1。

以上每步运行时间均为 $O(n)$，因此总运行时间也为 $O(n)$；第二步和第四步保证了排序算法的稳定性；额外空间开销包含第一步和第三步开销，为常数空间开销。

1-4

4. 从一个由 n 个互异元素构成的数组中选择第 i 个 $(i>1)$ 顺序统计量和找最小值的渐近运行时间相同。 ( )
Given an array A of n distinct elements, the asymptotic running time for selecting the ith element and selecting a minimum is the same.

True

SELECT 算法

1-5

5. T_1, T_2 是相同集合上的两棵二叉搜索树，若 T_1, T_2 的前序遍历序列相同，则两棵树相同。 ( )
Given two binary search trees T_1 and T_2 on the same set. If the inorder traversals of T_1 and T_2 are the same, they are the same tree.

False

如果集合上没有相同元素，则前序遍历可以唯一确定一颗二叉搜索树。但如果集合中存在相同元素，则不可。如 3,3 有两种画法

1-6

6. AVL 树是一种高度平衡的二叉搜索树：对于每一个结点 x，x 的左子树和右子树的高度差至多为 1。红黑树也是 AVL 树。 (　　)
An AVL tree is a binary search tree that is height balanced: for each node x, the heights of the left and right subtrees of x differ by at most 1. The red-black tree also is an AVL tree.

False

红黑树的左右子树高差有可能大于 1，所以红黑树不是严格意义上的平衡二叉树。

2-1

1. 我们在求解算法的时间复杂度时，通常假设：**过程调用中的参数传递**花费常量的时间，即使传递一个 N 个元素的数组也是如此。考虑这样一种参数传递策略：传递数组时，只复制过程可能访问的子区域，若数组 $A[p..q]$ 被传递，则时间为 $\Theta(q-p+1)$。请给出**归并排序**在该种参数传递策略下的**最坏情况**运行时间的递归式。
We assume that parameter passing during procedure calls takes constant time, even if an N-element array is being passed. We consider such a parameter-passing strategy: An array is passed by copying only the subrange that might be accessed by the called procedure. Time $= \Theta(q-p+1)$ if the subarray $A[p..q]$ is passed. Please give recurrences for the worst-case running times of merge-sort when arrays are passed using aforementioned method.

```
MERGE-SORT(A, p, r)
if p < r
    q = ⌊(p + r)/2⌋
    MERGE-SORT(A, p, q)
    MERGE-SORT(A, q + 1, r)
    MERGE(A, p, q, r)
```

时间复杂度计算公式依然是 $T(n) = 2T(\frac{n}{2}) + \Theta(n)$

2-2

2. 给定两个 n 位整数 X 和 Y。命题：计算 XY，我们需要 $\Omega(n^2)$ 次的一位整数的加法和乘法。请问该命题是否正确？如果不正确请给出你的答案。
Given two integers X, Y, each of n digits. Proposition: to compute the production XY, we always need to use $\Omega(n^2)$ additions and multiplications of one-digit integers. Is this proposition correct? If not, please give your answer.

不正确，通过分治法能将计算的时间复杂度降低到 $O(n^{\log 3})$

2-3

3. 对一个包含 n 个元素的集合，k 分位数是指能把有序集合分成 k 个等大小集合的第 $k-1$ 个顺序统计量。给出一个能找出某一集合的 k 分位数的 $O(n\lg k)$ 时间的算法。
The kth quantiles of an n-element set are the $k-1$ order statistics that divide the sorted set into k equal-sized sets (to within 1). Give an $O(n\lg k)$-time algorithm to find the kth quantiles of a set.

算法思想：分治策略，先找到 k 分位数中的第 $\lfloor k/2 \rfloor$ 位数，然后以此为主元将数组分为两部分，不断递归处理。

子算法：SELECT 算法，因为其时间复杂度为 $O(n)$。
时间复杂度分析：不妨假设 n 和 k 都为 2 的指数。用 $T(n)$ 表示问题规模为 n 时算法的运行时间，则可以写出递归式为 $T(n) = 2T(n/2) + cn$，c 为某个常数。且 $T(n/k) = O(1)$。则:

$$
\begin{aligned}
T(n) &\leq cn + 2T(n/2) \leq cn + 2c(n/2) + 4T(n/4) \\
&= 2cn + 4T(n/4) \leq 3cn + 8T(n/8) \\
&\dots \\
&\leq \log(k)cn + kT(n/k) \\
&\leq \log(k)cn + O(k) \\
&= O(n \log k)
\end{aligned}
$$

2-4

4. 对于图 1 所示的斐波那契堆，给出执行抽取最小结点操作之后的结果。
Please give the result of extracting the minimum node of the Fibonacci heap shown in Fig. 1.

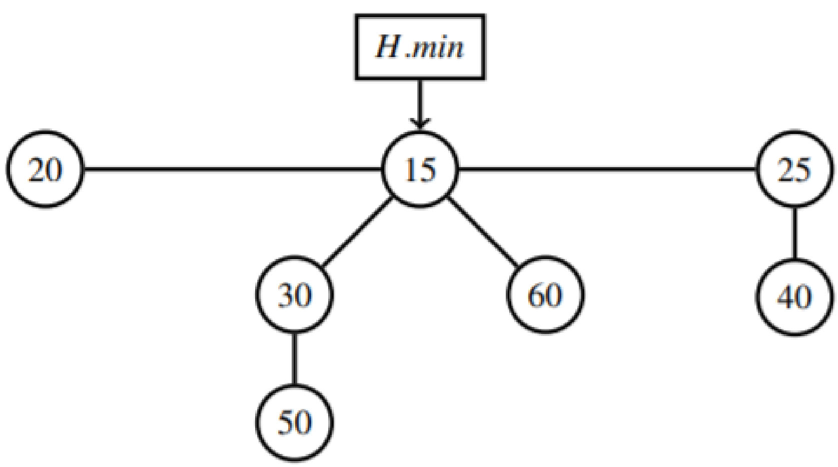


图 1: 斐波那契堆
Fig. 1 Fibonacci Heap

3-1

1. 排序和顺序统计量的计算在数据分析领域有着十分重要的作用，请回答下列问题:

(a) 给定一个包含 n 个互异的元素的集合，请简要描述如何在期望时间为 $\Theta(n)$ 的时间内找到第 k 小的元素。

(b) 设计一个算法，按顺序输出前 k 个最小的元素，简要描述该算法的思想并给出时间复杂度。要求该算法的时间复杂度小于 $\Theta(kn)$。

(c) 给定两个分别包含 n 个不同元素的有序序列 X 和 Y, 请设计一个 $\Theta(\lg n)$ 时间的算法，找到 X, Y 序列中所有元素的中位数。

(a) 随机选择划分元，将数组分为小于和大于划分元的两个部分，重复这一过程

RANDOMIZED-SELECT

```
RANDOMIZED-SELECT(A, p, r, i)
if p == r then
    return A[p]
q = RANDOMIZED-PARTITION(A, p, r)
k = q − p + 1
if i == k then
    return A[q] // the pivot value is the answer
if i < k then
    return RANDOMIZED-SELECT(A, p, q − 1, i)
else
    return RANDOMIZED-SELECT(A, q + 1, r, i − k)
```

(b) SELECT 算法依次输出所需要的元素

(c) 算法思路：将两个序列的中位数进行比较，每次删除这两个数中，较大数对应序列中比其中位数更大的部分以及较小数对应序列中比其中位数更小的部分。递归即可解决该问题

3-2

2. 一棵二叉搜索树 T 的前序遍历如下所示：

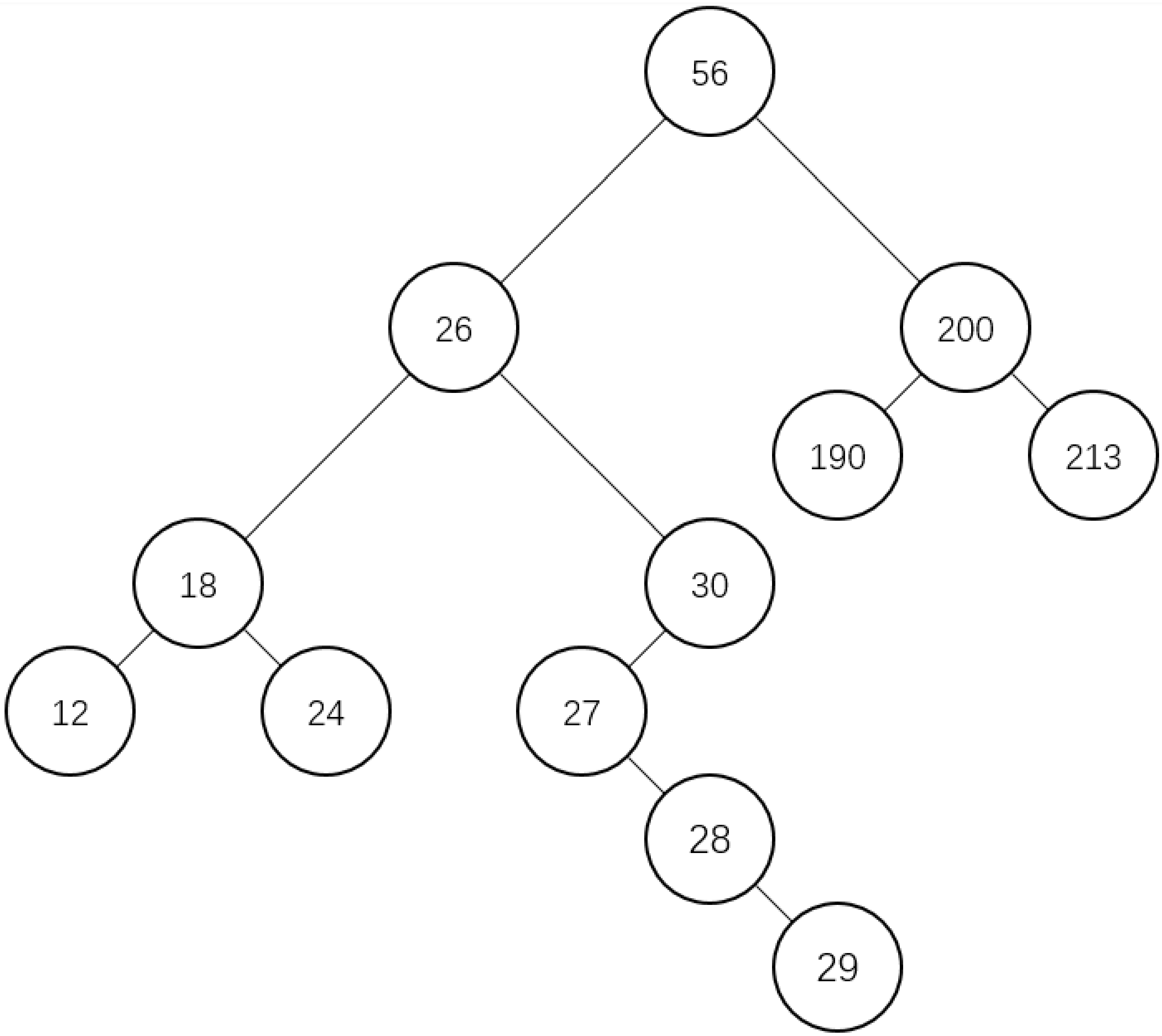
$$56, 26, 18, 12, 24, 30, 27, 28, 29, 200, 190, 213$$

(a) 画出一棵满足该前序遍历的二叉树 T。

(b) 是否可以给这棵树的结点着色使其成为一棵红黑树 T_1。若可以，画出 T_1，若不可以，说明理由。

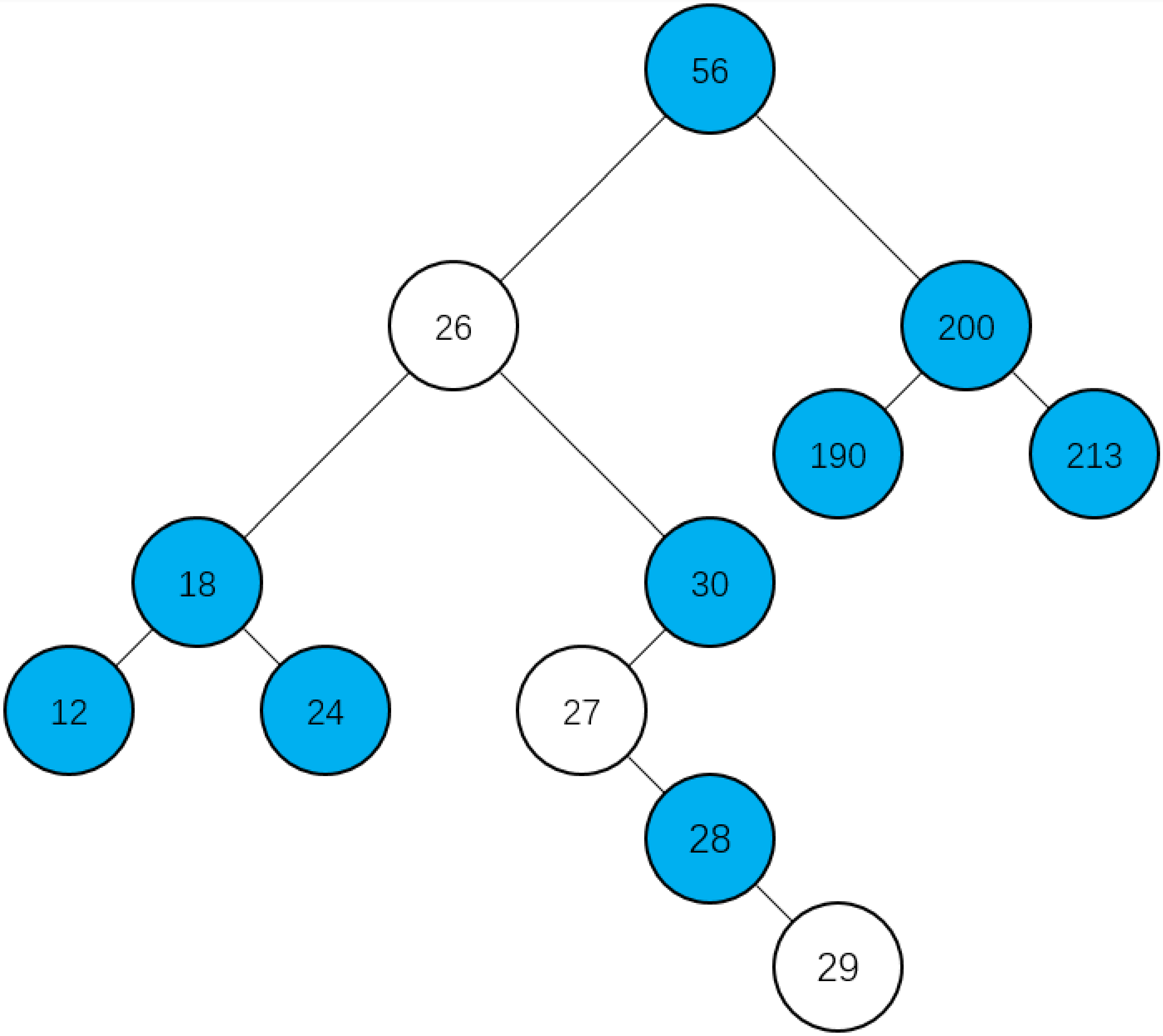
(c) 将结点 30 从 T 中删除得到 T_2，画出 T_2。

(a)

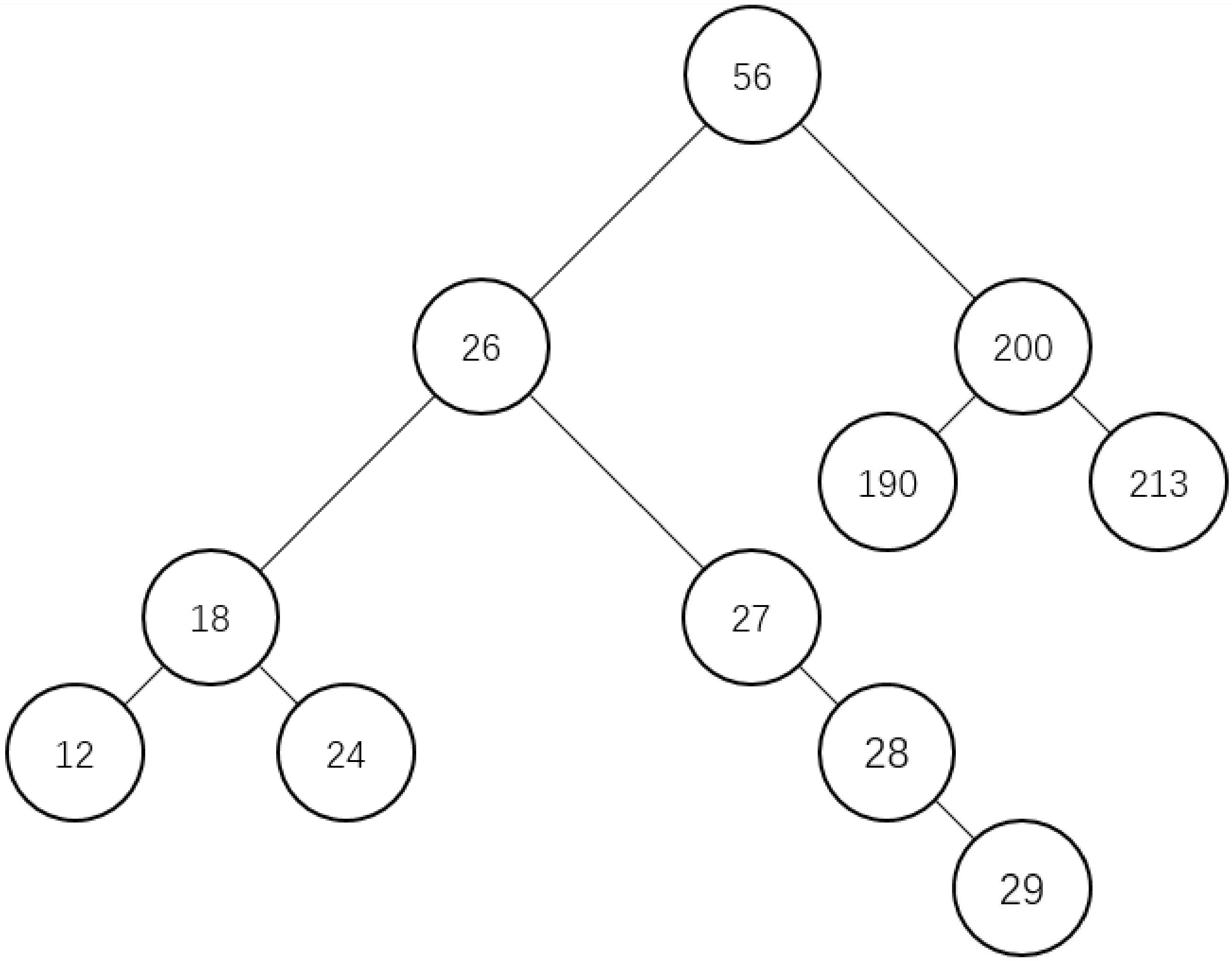


(b)

不可以 56-26-30-27-28-29 中至少要有 3 个黑，所以 200，,90，213 都为黑，所以 27一定为红，27左右子树黑色结点数量一定不同。



(c)



附加题

四、附加题（根据题目要求写出解答过程；每题 10 分，共 10 分）。

定义 Josephus 问题如下：假设 n 个人围成一个圆圈，给定一个正整数 m 且 $m \le n$。从某个指定的人开始，沿环将遇到的每第 m 个人移出队伍，每个人移出之后，继续沿环数剩下来的人。这个过程直到所有的 n 个人都被移出后结束。每个人移出的次序定义了一个来自整数 $1, 2, ..., n$ 的 $(n, m)-$Josephus 排列。例如，$(7, 3)-$Josephus 排列为 $(3, 6, 2, 7, 5, 1, 4)$。

(a) 假设 m 为一个常数，描述一个时间复杂度 $O(n)$ 的算法，使得对于给定的 n，能够输出 $(n, m)-$Josephus 排列。

(b) 假设 m 不是常数，简要描述时间复杂度为 $O(n\lg n)$ 的算法，使得对于给定的 n，能够输出 $(n, m)-$Josephus 排列。

(a) 创建一个双向循环链表。持续的在列表中前进 m 个位置，然后打印并删除当前节点，直到列表为空。 因为 m 是一个常数，我们最多推进到 mn 个地方，运行时间是 $O(n)$。

(b) 对给定的元素创建一个有序统计树，元素的排名从 0 开始到 $n-1$ 结束，并将每个元素的值存储到属性 x.value 中。打印排序为 $m-1$ 的元素，并将其从树中删除。并进行如下的操作：假定，第 k 次打印的元素排序为 r，则第 $k+1$ 次打印的元素排序为 $r-1+m \mod (n-k)$。由于在有序统计树中删除和查找需要 $O(\lg n)$ 的时间，在整个过程中我们要对 n 个元素执行上述操作，因此运行时间是 $O(n\lg n)$