2022 年秋季学期算法基础期中考试

学号 ________________　　姓名 ________________

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：

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

其中我们将 n/b 解释为 $\lfloor n/b \rfloor$ 或 $\lceil n/b \rceil$。那么 $T(n)$ 有如下渐进界：

1. 若对某个常数 $\varepsilon > 0$ 有 $f(n) = O(n^{\log_b a - \varepsilon})$，则 $T(n) = \Theta(n^{\log_b a})$。
2. 若对整数 $k \geq 0$ 有 $f(n) = \Theta(n^{\log_b a} \lg^k n)$，则 $T(n) = \Theta(n^{\log_b a} \lg^{k+1} n)$。
3. 若对某个常数 $\varepsilon > 0$ 有 $f(n) = \Omega(n^{\log_b a + \varepsilon})$，且对某个常数 $c < 1$ 和所有足够大的 n 有 $af(n/b) \leq cf(n)$，则 $T(n) = \Theta(f(n))$。

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 $T(n) = \Theta(f(n))$.

一、**判断题**（根据表述判断正误，并简要说明理由；每题 5 分，共 30 分; 每一题判断答案正确得 2 分，解释理由正确得 3 分。)。

1. (**T**,　**F**) 递归式 $T(n) = 4T(n/2) + n^2 + n\lg n + n$ 的解是 $T(n) = \Theta(n^2)$.
   The solution of the recurrence $T(n) = 4T(n/2) + n^2 + n\lg n + n$ is $T(n) = \Theta(n^2)$.

2. (**T**,　**F**) 堆排序和归并排序都是渐近最优的比较排序算法，他们在最坏情况下都需要做 $\Omega(n\log n)$ 次比较。
   Heapsort and merge sort are asymptotically optimal comparison sort, and both of them require $\Omega(n\log n)$ comparisons in the worst case.

3. (**T**, **F**) 给定 n 个 d 位数字, 其中每个数字最多可具有 k 个可能的值, 基数排序 (RADIX-SORT) 在 $\Theta(d(n \times k))$ 时间内正确地对这些数字进行排序。
Given n d-digit numbers in which each digit can take on up to k possible values, RADIX-SORT correctly sorts these numbers in $\Theta(d(n \times k))$ time.

4. (**T**, **F**) RANDOMIZED-SELECT 算法递归处理分区的两侧，算法的期望运行时间 (expected running time) 为 $\Theta(n)$。
RANDOMIZED-SELECT recursively processes both sides of the partition, and has an expected running time of $\Theta(n)$.

5. (**T**, **F**) 对于红黑树中的每一个结点 x, x 的左子树和右子树的高度差至多为 1。
For each node x in a red-black tree, the heights of the left and right subtrees of x differ by at most 1.

6. (**T**, **F**) 给定一棵二叉树的中序遍历和后序遍历，我们可以唯一确定一棵二叉树。
Given the inorder traversal and postorder traversal of a binary tree, we can determine a binary tree.

二、**简答题**（根据题目要求写出解答过程；每题 10 分，共 40 分）。

1. 对于递归式 $T(n) = 4T(n/2) + n^2 \lg^2 n$，给出 $T(n)$ 一个好的渐进上界.
   Give a good asymptotic upper bound for the recurrence$T(n) = 4T(n/2) + n^2 \lg^2 n$.

2. 解释为什么桶排序在最坏情况下运行时间是 $\Theta(n^2)$？我们应该如何修改算法，使其在保持平均情况为线性时间代价的同时，最坏情况下时间代价为 $O(nlgn)$。

   Explain why the worst-case running time for bucket sort is $\Theta(n^2)$. What simple change to the algorithm preserves its linear average-case running time and makes its worst-case running time $O(nlgn)$?

3. 证明：对于任一包含 n 个元素的堆中，至多有 $\lfloor n/2^{k+1} \rfloor$ 个高度为 h 的结点。

   Show that there are at most $\lfloor n/2^{k+1} \rfloor$ nodes of height h in any n-element heap.

4. 对于图一所给的斐波那契堆, 黑色节点表示结点的 mark 属性为 true:
   1) 把关键字 50 减为 5, 画出操作后的最后的结果 (需考虑级联操作)。
   2) 在 1）的基础上，抽取最小结点 (需要考虑 CONSOLIDATE 操作), 画出操作后最后的结果。

   Given a Fibonacci heap below. the mark attribute of black node is true:
   1)Please give the result of decrease the value of key 50 down to 5.
   2)Please give the result of extracting the minimum node, based on the result of 1) (CONSOLIDATE should be considered).

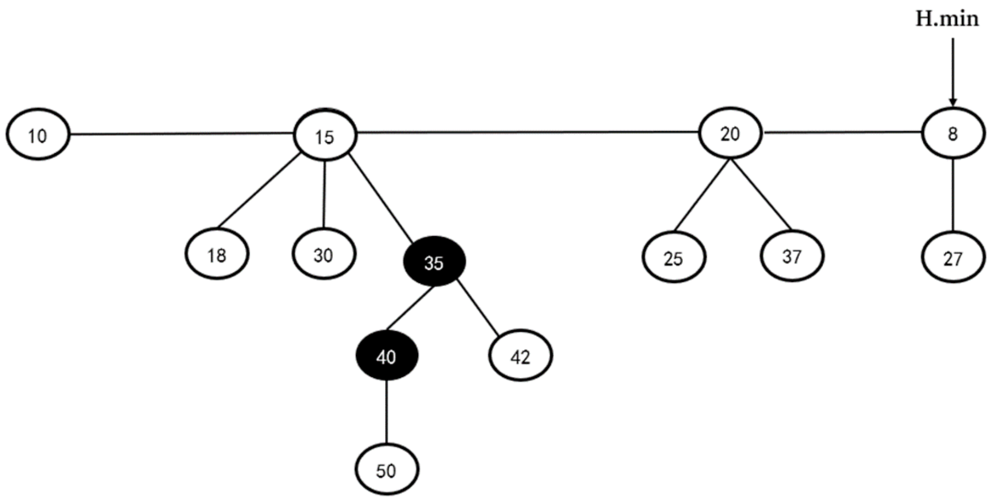

图 1: 斐波那契堆
Fig. 1 Fibonacci Heap

三、**综合题**（根据题目要求写出解答过程；每题 15 分，共 30 分）。

1. 要在 n 个数中选出第 i 个顺序统计量，SELECT 在最坏情况下需要的比较次数 $T(n)$ 满足 $T(n)=\Theta(n)$。但是，隐含在 Θ 记号中的常数项是非常大的。当 i 相对 n 来说很小时，我们可以实现一个不同的算法，它以 SELECT 作为子程序，但在最坏情况下所做的比较次数更少。
(a) 设计一个能用 $U_i(n)$ 次比较在 n 个元素中找出第 i 小元素的算法，其中，

$$U_i(n)=\begin{cases} T(n), & i \geq n/2 \\ \lfloor n/2 \rfloor + U_i(\lceil n/2 \rceil)+T(2i), & otherwise \end{cases}$$

(b) 证明：如果 $i<n/2$, 则有 $U_i(n)=n+O(T(2i)\lg(n/i))$。
(c) 证明：如果 i 是小于 $n/2$ 的常数，则有 $U_i(n)=n+O(\lg n)$。
(d) 证明：如果对所有 $k \geq 2$ 有 $i=n/k$, 则 $U_i(n)=n+O(T(2n/k)\lg k)$。

We showed that the worst-case number $T(n)$ of comparisons used by SELECT to select the ith order statistic from n numbers satisfies $T(n)=\Theta(n)$, but the constant hidden by the Θ-notation is rather large. When i is small relative to n, we can implement a different procedure that uses SELECT as a subroutine but makes fewer comparisons in the worst case.
(a) Describe an algorithm that uses $U_i(n)$ comparisons to find the i-th smallest of n elements, where,

$$U_i(n)=\begin{cases} T(n), & i \geq n/2 \\ \lfloor n/2 \rfloor + U_i(\lceil n/2 \rceil)+T(2i), & otherwise \end{cases}$$

(b)Show that, if $i<n/2$, then $U_i(n)=n+O(T(2i)\lg(n/i))$。
(c)Show that, if i is a constant less than $n/2$, then $U_i(n)=n+O(\lg n)$。
(d)Show that, if $i=n/k$ for $k \geq 2$, then $U_i(n)=n+O(T(2n/k)\lg k)$。

2. 定义一棵二叉树 T 的**路径总长度** (total path length) $P(T)$ 为 T 中所有结点 x 的深度之和，将每个结点 x 的深度表示为 $d(x,T)$。
(a) 证明：T 中的一个结点平均深度是

$$\frac{1}{n}\sum_{x\in T} d(x,T) = \frac{1}{n}P(T)$$

(b) 设 $P(n)$ 表示有 n 个结点的随机构建二叉搜索树的平均路径总长度，证明：

$$P(n) = \frac{1}{n}\sum_{i=0}^{n-1}(P(i)+P(n-i-1)+n-1)$$

(c) 请给出快速排序的一种实现，使快速排序中对一组元素的比较与将这些元素插入一棵二叉搜索树中所需的比较恰好相同。(这些比较的次序可以不同，但出现的比较一定要一样。)

We define the *total path length* $P(T)$ of a binary tree T as the sum, over all nodes x in T, of the depth of node x, which we denoted by $d(x,T)$.

(a) Argue that the average depth of a node in T is $\frac{1}{n}\sum_{x\in T} d(x,T) = \frac{1}{n}P(T)$.
(b) Let $P(n)$ denote the average total path length of a randomly built binary search tree with n nodes. Show that

$$P(n) = \frac{1}{n}\sum_{i=0}^{n-1}(P(i)+P(n-i-1)+n-1)$$

(c) Describe an implementation of quicksort in which the comparisons to sort a set of elements are exactly the same as the comparisons to insert the elements into a binary search tree. (The order in which comparisons are made may differ, but the same comparisons must occur.)

四、附加题（根据题目要求写出解答过程，共 10 分）。
与快速排序中的分析一样，我们假设所有的元素都是互异的，输入数组 A 中的元素被重命名为 $z_1, z_2, ..., z_n$，其中 z_i 是第 i 小的元素。因此，调用 RANDOMIZED-SELECT$(A, 1, n, k)$ 返回 z_k。

对所有 $1 \le i < j \le n$, 设

$$X_{ijk} = I\{\text{在执行算法查找}z_k\text{期间，}z_i\text{与}z_j\text{进行过比较}\}$$

1. 给出 $\mathbb{E}[X_{ijk}]$ 的准确表达式。（提示：你的表达式可能有不同的值，依赖于 i、j、k 的值。）
2. 设 X_k 表示在找到 z_k 时 A 中元素的总比较次数，证明：

$$\mathbb{E}[X_k] \le 2\left(\sum_{i=1}^{k}\sum_{j=k}^{n}\frac{1}{j-i+1} + \sum_{j=k+1}^{n}\frac{j-k-1}{j-k+1} + \sum_{i=1}^{k-2}\frac{k-i-1}{k-i+1}\right)$$

3. 证明：$\mathbb{E}[X_k] \le 4n$，并说明 RANDOMIZED-SELECT 的期望运行时间是 $O(n)$。

As in the quicksort analysis, we assume that all elements are distinct, and we rename the elements of the input array A as $z_1, z_2, ..., z_n$, where z_i is the i th smallest element. Thus, the call RANDOMIZED-SELECT$(A, 1, n, k)$ returns z_k.

For $1 \le i < j \le n$,

$$X_{ijk} = I\{z_i \text{is compared with} z_j \text{sometime during the execution of the algorithm to find } z_k\}$$

1. Give an exact expression for $\mathbb{E}[X_{ijk}]$ (Hint: Your expression may have different values, depending on the values of i、j and k.)
2. Let X_k denote the total number of comparisons between elements of array A when finding z_k. Show that

$$\mathbb{E}[X_k] \le 2\left(\sum_{i=1}^{k}\sum_{j=k}^{n}\frac{1}{j-i+1} + \sum_{j=k+1}^{n}\frac{j-k-1}{j-k+1} + \sum_{i=1}^{k-2}\frac{k-i-1}{k-i+1}\right)$$

3. show that $\mathbb{E}[X_k] \le 4n$. Conclude that, assuming all elements of array A are distinct, RANDOMIZED-SELECT runs in expected time $O(n)$.

附：RANDOMIZED-SELECT 返回数组 $A[p..r]$ 中第 i 小的元素

```
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]
else if i<k then
    return RANDOMIZED-SELECT(A, p, q − 1, i)
else
    return RANDOMIZED-SELECT(A, q + 1, r, i − k)
```