

- (1) 只准讨论思路，严禁抄袭，查重时发现重合度极高的实验代码取消当次实验分数。
- (2) 只能阅读 bb 上的材料和教材算法导论。严禁网上搜寻任何材料，答案或者帮助。
- (3) 禁止直接使用库函数实现实验题目的主要算法。

问题 1 (40 分). 在实际应用中，根据不同操作的特点，我们可以组合基础的数据结构，以得到更好的性能。

下面我们将尝试把 Hash 和 Heap 组合起来，维护一个成绩列表，其中每个元素由名字和成绩组成。名字是大写字母构成的字符串，成绩是一个 1 到 100 之间的整数。

该数据结构需要支持下列操作（ n 为当前堆的大小， k 是成绩最高元素的个数）：

1. 插入一个元素，名字为 Name，成绩为 Score，时间复杂度为 $O(\log n)$
2. 删除名字为 Name 的元素，时间复杂度为 $O(\log n)$
3. 查询名字为 Name 的元素的的成绩，无此元素则返回 0，时间复杂度为 $O(1)$
4. 查询当前成绩最高的元素，输出元素的名字，如有多个最高成绩，名字按字典序输出，时间复杂度为 $O(k)$

问题 2 (40 分). 维护一个红黑树，每个元素 $a_i \in \mathbb{N}$ 支持下列操作：

1. 插入元素
2. 删除元素
3. 查找当前树中第 k 大元素
4. 查询 $[A, \infty)$ 区间中的最小元素
5. 查询 $[L, R]$ 区间中的元素个数

问题 3 (20 分). 给定一个序列 a_1, \dots, a_n ，设计一个算法来计算最长严格递增子序列 s_1, \dots, s_m 的长度。