作业要求：说明思路与符号，清晰简洁的伪代码，必要的时间复杂度分析和必要的正确性分析。可以直接调用基本的数据库和已讨论过的算法/程序（如排序、找中位数、二分查找等）。

问题 1 (30 分). 在课上我们观察到了 Shell Sort 的一个重要性质：
一旦对数组 A 调用了步长为 h 的选择排序，A 数组在 Shell Sort 的未来时刻总是有 (注：此处的未来时刻指调用 $\textsc{InsertionSort}(h')$ 之后，不包括 $\textsc{InsertionSort}(h')$ 的进行过程中):

$$1 \le i \le n-h, A[i] \le A[i+h] \tag{1}$$

实际上，只需要对任意步长 h_1 和 h_2，证明先调用 $\textsc{InsertionSort}(h_2)$ 后调用 $\textsc{InsertionSort}(h_1)$，依然有 $\forall i : A[i] \le A[i+h_2]$；然后归纳即可。

(a) 对任意的 h_1 和 h_2 严格证明以上性质 [15 分]。

(b) 对 Lecture 2 中 ShellSort 部分的 Lemma 1（见 bb.ustc.edu.cn），利用课堂上的框架，提供一个完整的严格证明 [15 分]。

问题 2 (30 分). (a) 令 $T(n)$ 为 n 个元素的快速排序期望时间。对课件中的 $\textsc{QuickSort}$ 代码，写出 $T(n)$ 的递归式。

(b) 证明递归式 $T(n) = O(n \log n)$。

(c) 基于课件的 $\textsc{QuickSort}$ 代码，设计期望时间复杂度为 $O(n)$ 的寻找第 k 号数的算法（从小到大）。

(d) 计算每对元素的比较概率证明其期望时间。

问题 3 (20 分). 参考算法导论第 8.1 章，考虑 σ 为 $1, \ldots, n$ 所有 $n!$ 个排列中均匀随机产生的。

(a) 对任何基于比较排序的确定性算法 A，证明对至少 99% 的排列 σ，A 的运行时间为 $\Omega(n \log n)$。也就是

$$\Pr_{\sigma}\left[Time\Big(A(\sigma) \Big) \ge 0.5 n \log n \right] \ge 0.99.$$

其中 $Time(A(\sigma))$ 表示算法 A 对输入 σ 的运行时间。

(b) 考虑任一基于比较排序的随机化算法 $A(\sigma, r)$（永远输出正确答案，但运行时间为随机变量；其中 σ 为输入串 r 为随机种子），证明对 50% 的排列 σ 有：

$$\mathop{\mathbb{E}}_{r}\left[Time\Big(A(\sigma, r) \Big) \right] \ge 0.4 n \log n.$$

提示：假定算法 A 采用固定长度 $l(n)$ 的随机数 r，即 $r \in \{0,1\}^{l(n)}$。固定随机数 r 后，$A(\cdot, r)$ 为确定性算法。

问题 4 (20 分)**.** 事实上，RadixSort 能够对有限范围的分数进行排序。给定 n 个分数 a_i/b_i 其中 a_i 和 b_i 皆为 $\{1, 2, \ldots, n^2\}$ 间的正整数，设计时间 $O(n)$ 的算法对分数 a_i/b_i 排序。