互评题 1：

给定数轴 X 上 n 个不同点的集合$\{x_1, x_2, \ldots, x_n\}$，其中$x_1 < x_2 < \cdots < x_n$。现在用若干个长度为 1 的闭区间来覆盖这些点。设计一个算法找到最少的闭区间个数和位置。

要求如下：

1. 请简要叙述贪心算法的策略；（1 分）
2. 请证明你的贪心策略能够给出最优解（2 分）；
3. 请分析算法的空间和时间复杂度（2 分）。

参考答案：

（注：这里给出的是 O(n)复杂度的算法，如果给出其他算法但可以正确求出最优解，也可酌情给分）

1. 贪心策略：从x_1取起，第一个区间是$[x_1, x_1+1]$。顺序考察后面的点，假设最后一个落入该区间的点是x_k，即$x_k \le x_1 + 1, x_{k+1} > x_1 + 1$。那么下一个区间是$[x_{k+1}, x_{k+1}+1]$。按照这样的办法直到剩下的所有点都落入最后一个区间为止。
2. 正确性证明：对任何正整数 j，算法进行到第 j 步，得到第 1 到 j 个区间，一定存在一个最优解包含这 j 个区间。对 j 归纳，下面使用符号$\tau(a)$表示单位区间$[a, a+1]$。

   $j = 1$，即证明存在最优解包含了区间$\tau(x_1)$。

   x_1点一定在任何最优解的某个区间中，而且在第一个区间里。若不然，在包含x_1的区间左边还有别的区间，那么这些区间不可能包含 X 中的点，因此这些区间可以从解中去掉，与解的最有型矛盾。设最优解的第一个区间是$\tau(a)$，假设$a < x_1$，那么用$\tau(x_1)$替换$\tau(a)$，$\tau(x_1)$完全覆盖了$\tau(a)$中的点，因此替换后仍旧是最优解。

   $j = k$，假设存在最优解 T 包含算法钱 k 步选择的 k 各区间，即

   $$T = \{\tau(x_{i_1}), \tau(x_{i_2}), \ldots, \tau(x_{i_k})\} \cup T'$$

   其中$x_{i_1}, x_{i_2}, \ldots, x_{i_k}$是算法从 X 中选择的 k 各区间的做端点，且$i_1 = 1$。设这 k 个区间包含了点$x_1, x_2, \ldots, x_p$，那么考虑后面$n-p$个点$x_{p+1}, \ldots, x_n$的区间选择子问题，而 T'是子问题$X' = \{x_{p+1}, \ldots, x_n\}$的一个最优解。如若不然，X'有更好的解 T''，那么用 T''替换 T'，就可以得到原问题的一个比 T 更好的解，这与归纳假设矛盾。考虑子问题 X'，有归纳基础，对于 X'存在一个最优解$T_1 = \{\tau(x_{p+1})\} \cup T_2$，下面证明

$$T^* = \{\tau(x_{i_1}), \tau(x_{i_2}), \dots, \tau(x_{i_k})\} \cup T_1$$

也是原问题的最优解。假设T^*不是最优解，那么T_1比 T′的区间个数至少多 1，因此与T_1是$X' = \{x_{p+1}, \dots, x_n\}$的最优解矛盾，于是得到

$$\begin{aligned} T^* &= \{\tau(x_{i_1}), \tau(x_{i_2}), \dots, \tau(x_{i_k})\} \cup \{\tau(x_{p+1})\} \cup T_2 \\ &= \{\tau(x_{i_1}), \tau(x_{i_2}), \dots, \tau(x_{i_k}), \tau(x_{p+1})\} \cup T_2 \\ &= \{\tau(x_{i_1}), \tau(x_{i_2}), \dots, \tau(x_{i_k}), \tau(x_{i_{k+1}})\} \cup T_2 \end{aligned}$$

从而算法前 k+1 步的选择也可以导致最优解。根据数学归纳法，命题得证。

3. 算法最坏情况下的时间复杂度是$O(n)$，空间复杂度是$O(n)$，这里如果考虑输出是覆盖区间的起始位置的数组，空间复杂度是$O(n)$。如果每次计算出一个位置，马上就输出这个位置信息，而不把得到的位置信息存在输出数组里，那么空间复杂度是$O(1)$。两个结果都对。大家判题时对这两种结果都应该给分。。

评分标准：

1. 策略正确 1 分，错误 0 分
2. 证明过程正确给 2 分，过程有漏洞但思路正确给 1 分，证明错误给 0 分
3. 时间和空间复杂度各占 1 分

互评题 2：

一个公司需要购买 n 个密码软件的许可证，按规定每个月至多可得到一个软件许可证. 每个许可证当前售价都是 1000 元，但是第 i 个许可证的售价将按照 r_i >1 的指数因子增长，i=1,2,…,n.　例如，第 i 个许可证的售价在 1 个月后将是 r_i×1000 元，2 个月后将是 r_i^2×1000 元，k 个月后将是 r_i^k ×1000 元. 假设 $r_1, r_2, \dots, r_n$ 是给定正整数，试给出一个购买许可证的顺序，以使得花费的总钱数最少.

1. 令问题的解 I 是 $i_1,i_2,\dots,i_n$，其中 $i_1,i_2,\dots,i_n$ 是 1,2,…,n 的排列。问：按照这个解给出的顺序，当月购买第一个许可证，以后每个月恰好购买一个软件许可证，总的花费 $V(I)$是多少元？　(1 分)
2. 设计一个贪心算法求解这个问题，用一段简短的话说明该算法的贪心策略.(1 分)
3. 证明算法的正确性. （2 分）
4. 求出算法在最坏情况下的时间复杂度. （1 分）

参考答案：

1. 总花费

$$V(I)=1000\times(1+r_{i_2}+r_{i_3}{}^{2}+\ldots+r_{i_{n-1}}{}^{n-2}+r_{i_n}{}^{n-1})$$

2. 按照 r_i 从大到小的顺序重新排列软件许可证的编号，即使得 $r_1 \geq r_2 \geq \ldots \geq r_n$，然后依照顺序 1,2,…,$n$ 购买.
3. 证明

设最优解为 OPT(I)，假设在 OPT(I)存在逆序，即 $r_j < r_i$，但是 j 在 i 前面购买. 一定有相邻的逆序, 即存在 i 和 j, 使得 j 在 i 前面与 i 相邻. 设 j 是第 t 个月后购买的，i 是第 $t+1$ 个月后购买的. 交换 i 与 j 得到解 $S(I)$，那么花费之差

$$\begin{aligned}&V(\mathrm{OPT}(I))-V(S(I))\\&=(r_j^t\times 1000+r_i^{t+1}\times 1000)-(r_i^t\times 1000+r_j^{t+1}\times 1000)\\&=[\,r_i^t(r_i-1)-r_j^t(r_j-1)\,]\times 1000\end{aligned}$$

由于 $r_i > r_j$，于是 $r_i-1 > r_j-1$ 且 $r_i^{\,t} > r_j^t$，得到上式 > 0.

交换减少了逆序，而总花费也减少. 至多存在 $n(n\text{-}1)/2$ 个逆序，经过有限次交换，就得到算法的解. 算法解的总花费达到最小.

4. 时间复杂度为 $O(n\log n)$.

评分标准：

1. 做对了给 1 分，错了给 0 分。
2. 购买顺序是按 ri 递减顺序的给 1 分，否则给 0 分
3. 采用交换论证方法证明的：

   说明了交换的规则，交换次数有限，交换后花费减少，可以给 1 分。

   列出交换后花费会减少的计算公式，公式正确的给 1 分，不正确的给 0 分。

   采用对步数归纳证明方法的：

   命题叙述正确和归纳基础证明正确的给 1 分

   归纳步骤证明正确的给 1 分
4. 时间复杂度正确的给 1 分，不正确的 0 分。