除明确要求外，本次作业只需简要描述算法实现思路及过程并分析最坏时间复杂度，不需要写出（伪）代码，也不需要证明正确性。

Q1.（$25 + (10 + 10) + 10 = 55$ 分）

给定整数序列 $\mathbf{x}$，如果 x_i 有一个恰好比它小 1 的相邻元素，即 $x_{i-1} = x_i - 1$ 或 $x_{i+1} = x_i - 1$，那么就可以将 x_i 在序列中删去（只删 x_i 自己，不删其他位置上等于 x_i 的元素）。我们想知道：这样的操作最多能够进行多少次？

一种基于贪心选择的思路是：尽可能保留较小的元素，故每次都删去当前最大的可删除元素（若有多个，任选其中一个）

(1) 选择合适的数据结构，对这一思路进行实现，写出伪代码（或者直接给出用某种编程语言实现的代码）并分析最坏时间复杂度。本题得分会参考你给出的实现的最坏时间复杂度。

(*2) 贪心算法设计的一大难点是：算法正确性的证明往往并不简单。以下证明采用了“决策包容性”的手法。

(*2.1) 记 x_i 是 $\mathbf{x}$ 中当前能够删去的最大元素（之一）。证明：对于任意 $x_j = x_i + 1, |j - i| > 1$，无论接下来如何操作 $\mathbf{x}$（操作序列可以是任意的，不一定由贪心算法给出），x_i 与 x_j 都不可能相邻。

(*2.2) 记 x_i 是 $\mathbf{x}$ 中当前能够删去的最大元素（之一），记：

$$D = \{\text{所有合法删除序列}\}$$
$$D' = \{\text{以}x_i\text{起始的合法删除序列}\}$$

证明：存在 $\varphi : D \to D'$，使得 $\forall d \in D$，$\varphi(d)$ 不短于 d

这说明最长的删除序列一定能够在 D' 中取到，即贪心算法正确。

(*3) 若已知所有元素的取值均为 $1 \sim m$，试设计一个最坏时间复杂度为 $O(m + n)$ 的算法。

Q2.（$15+10+15+10=50$ 分）

为防止学生沉迷游戏，某大学严禁计算机专业大一学生在宿舍内使用电脑。宿舍楼某层一共有 n 间宿舍，这些宿舍呈直线排列。初始时，第 i 间宿舍内私藏了 x_i 台电脑。从第一天晚上开始，每间宿舍都会有一名同学将一台新买的电脑藏到宿舍中。

作为一名铁面无私的宿舍管理员，你开展了为期 k 天的“缴尽脑汁”行动：第一天早上，你进入某一个宿舍，将其内私藏的电脑全部缴获（数量为某个 x_i）；第二天早上，你可以进入一间相邻的宿舍进行缴获（因为第一天晚上他们又藏了一台电脑，因此数量为 $x_{i\pm1}+1$ 之一），或者留在原地（此时你只能缴获到一台电脑）；以此类推，每天你都可以进入相邻宿舍或留在原地。

(1) 若 $k \le n$，试设计一个具有 $O(n)$ 最坏时间复杂度的算法，以计算你能缴获的电脑数量的最大值。

(2) 证明 (1) 中算法的正确性。

提示：确定缴获数量的一个上界，并证明你的策略达到了此上界。

(*3) 若 $k > n$，试设计一个具有 $O(n)$ 最坏时间复杂度的算法，以计算你能缴获的电脑数量的最大值。

(*4) 证明 (3) 中算法的正确性。

提示：确定未缴获数量的一个下界，并证明你的策略达到了此下界。

Q3.（$10+15+(15+20)=60$ 分）

排列优化问题是一类重要的优化问题。给定集合 X 以及目标函数 $f : X^n \to \mathbb{R}$，一个排列优化问题指的是：对于输入序列 $\mathbf{x} \in X^n$，求 $\underset{\mathbf{y}\text{是}\mathbf{x}\text{的排列}}{\arg\max} f(\mathbf{y})$。若存在多个排列都能取得最大值，只需求出其中一个。

许多排列优化问题可以用基于贪心的元素对换进行分析，相应地，这些问题的解往往呈现某种有序结构。

(1) 若 $X=\mathbb{N}$, $f(\mathbf{y})=\sum_{i=1}^{n} i \cdot y_i$, 设计一个最坏时间复杂度为 $O(n \lg n)$ 的算法解决排列优化问题。

(2) 若 $X=\mathbb{N}$，$f(\mathbf{y})=\sum_{i=1}^{n} \max\{0, y_i - i\}$，设计一个最坏时间复杂度为 $O(n \lg n)$ 的算法解决排列优化问题。

(*3) 我们有以下结论：

如果存在 X 上的全预序 $\lesssim$，使得：任意 $\mathbf{y} \in X^n$，记 $\mathbf{y}'$ 是将 $\mathbf{y}$ 中 y_i, y_{i+1} 交换得到的排列，$y_{i+1} \lesssim y_i$ 是 $f(\mathbf{y}') \geq f(\mathbf{y})$ 的充分条件，则排列优化问题的最大值可以在任意一个 $\mathbf{z}$ 处取到，其中 $\mathbf{z}$ 是 $\mathbf{x}$ 的排列且满足 $z_1 \lesssim z_2 \lesssim \cdots \lesssim z_n$

(*3.1) 证明这一结论的正确性，并用这一结论证明你在 (1)、(2) 中设计的算法的正确性。

(*3.2) “双工序调度问题” 定义如下：有 n 个待加工的工件，每个工件必须先完成第一步加工后才能进行第二步加工，第 i 个工件进行两步加工的用时分别为 α_i, β_i。因为机器数量有限，因此每个步骤同时只能有一个工件在加工。要求确定工件的加工顺序，使得总用时最短。

这一问题可以形式化地表述为：令 $X=\mathbb{R}^+ \times \mathbb{R}^+$，记 $x_i=\langle \alpha_i, \beta_i \rangle, y_i=\langle a_i, b_i \rangle$，定义完成 i 个工件所需的时间：

$$t_i(\mathbf{y})=\begin{cases} a_1+b_1 & ,i=1 \\ \max\left\{t_{i-1}(\mathbf{y}), \sum_{j=1}^{i} a_j\right\}+b_i & ,2 \leq i \leq n \end{cases}$$

试设计最坏时间复杂度为 $O(n \lg n)$ 的算法，求出 $\underset{\mathbf{y}\text{是}\mathbf{x}\text{的排列}}{\arg\min}\ t_n(\mathbf{y})$，你需要对算法正确性给出证明。

提示：验证 $\min\{a_{i+1}, b_i\} \leq \min\{a_i, b_{i+1}\}$ 是 $t_n(\mathbf{y}') \leq t_n(\mathbf{y})$ 的充分条件，并将其“改造”为一个全预序。

Q4. ($((5+5+5+10)+10=35$ 分)

考虑一个仅由左右括号及星号组成的字符串 s，我们希望判断能否通过将星号替换为括号或空，使得整个字符串的括号是正确匹配的。正确匹配的例子如：空字符串、()、(())、()()、(()()) 等。

对于字符串 $s_1=$"$\star(\star()$"，可以将第一个星号替换为空，将第二个星号替换为"$)$"，从而得到正确匹配的字符串"()()"；但对于 $s_2=$"$(\star((\star$"，无论如何替换，都无法使得括号正确匹配。

(1) 对于这一问题，以下给出一个基于贪心匹配的算法：

首先，从左向右扫描每个字符，当扫描到右括号时：

- 若左侧还有剩余的左括号，将其中最右边的左括号与当前右括号一并删除
- 若左侧没有剩余的左括号，但有剩余的星号，将其中最右边的星号与当前右括号一并删除
- 若左侧为空，算法结束，输出`false`

在第一遍扫描的基础上从右向左扫描每个字符：

- 当扫描到左括号时，算法结束，输出`false`
- 当扫描到星号时，若左侧还有剩余的左括号，将其中最右边的左括号与当前星号一并删除

算法过程中，若任意时刻字符串中既没有左括号也没有右括号，则算法结束，输出`true`

(1.1) 举例说明第一遍扫描中"优先匹配左括号，没有左括号才匹配星号"这一顺序不能调换。

(1.2) 举例说明第一遍扫描中必须匹配"左侧最右边的左括号"，而不是"左侧最左边的左括号"。

(1.3) 举例说明第二遍扫描中必须匹配"左侧最右边的左括号"，而不是"左侧最左边的左括号"。

(1.4) 试使用合适的数据结构，实现这一算法并分析最坏时间复杂度。本题得分会参考你给出的实现的最坏时间复杂度。

(*2) 证明 (1) 中算法的正确性。