Задача A. Сумма цифр

Заметим, что итоговые значения в массиве после применения всех операций будут: $1, 2, 3, \ldots, 9, 1, 2, 3, \ldots 9, \ldots$. Тогда чтобы найти сумму на отрезке достаточно посчитать количество полных блоков из чисел от 1 до 9, и обработать края.

Задача B. Множество

Отсортируем массив a. Оптимальная выборка — подотрезок отсортированного массива. Для отрезка $[l, r]$ длины $m = r - l + 1$ сумма всех попарных разностей равна

$$\sum_{i=l}^{r} a_i \cdot (2(i - l + 1) - m - 1).$$

Перепишем через префиксные суммы $P_1[i] = \sum a_i$ и $P_2[i] = \sum i \cdot a_i$:

$$F(l, r) = -(m+1)(P_1[r] - P_1[l-1]) + 2\big(P_2[r] - P_2[l-1] - (l-1)(P_1[r] - P_1[l-1])\big).$$

Теперь достаточно использовать бинарный поиск по ответу, и проверять что сумма на заданном подотрезке меньше или равна данной.

Задача C. Турнир

Лемма. Если игрок x может выиграть турнир, а игрок y может победить x, то y тоже может выиграть турнир.

Пусть S — множество игроков, способных выиграть турнир.

Пусть T — минимальное непустое множество игроков такое, что ни один игрок, не входящий в T, не может победить игрока из T.

Докажем что $S = T$

$S \subseteq T$**:** По определению T, ни один игрок вне T не может выиграть турнир. Из утверждения 1 следует, что ни один игрок вне S не может победить игрока из S. По определению T имеем $|T| \leqslant |S|$.

$T \subseteq S$**:** Предположим, что существует игрок $x \in T$, но $x \notin S$. Так как $|T| \leqslant |S|$, существует игрок $y \in S$, но $y \notin T$. По определению T, y не может победить x, значит, x побеждает y. Но тогда по лемме выше, $x \in S$ — противоречие.

Из $S \subseteq T$ и $T \subseteq S$ получаем $S = T$.

Введём следующие обозначения:

- Пусть $M(p)$ — максимальный ранг игрока p среди всех кортов.
- Пусть $C(k)$ — количество игроков p, у которых $M(p) \leqslant k$.
- Легко видеть, что $C(k) \leqslant k$.

Пусть t — минимальное число, для которого $C(t) = t$. Тогда T — это множество всех игроков p, у которых $M(p) \leqslant t$.

Введём $D(k) = k - C(k)$. Тогда t это первый 0 в массиве D.

Каждое обновление изменяет только два значения $M(p)$, поэтому мы можем использовать дерево отрезков для эффективного управления этими обновлениями.

Задача D. Новая застройка

Для каждого индекса i определим множество кандидатов:

$$C_i = \{a_j - 1, a_j, a_j + 1 \mid i - 4 \leqslant j \leqslant i + 4\}.$$

Состояние ДП:

$dp[i][y][p]$ = минимальная стоимость решения для префикса i, где $b_i = y$ и b_i последний знак соседних элементов не считая равенств это p и $y \in C_i$.

Для переходов нужно просто перебрать следующее значение столбца.

Докажем, что такого C_i достаточно. Для этого нужно доказать, что не бывает плато длины больше 4

- **Интервал** $[\ell, r]$ — множество индексов $\ell, \ell+1, \ldots, r$.
- Интервал называется **плохим**, если для всех i в нём $b_i \neq a_i$.
- **Плато** — максимальный интервал одинаковых значений b_i.
- Плато называется **пиком**, если $b_{\ell-1} < b_\ell = \cdots = b_r > b_{r+1}$ (крайние случаи считаются тривиальными пиками/впадинами), и **впадиной** наоборот.

1. Внутри пика $[\ell, r]$ (не на границе) для любого $i \in (\ell, r)$ выполнено $a_i \geqslant b_i$. Иначе можно уменьшить b_i до a_i, улучшив стоимость и сохранив пикообразность.
2. Не может быть плохого интервала длины 4 строго внутри пика. Иначе можно модифицировать значения, не увеличивая стоимость, но увеличив число совпадений $a_i = b_i$. Для этого выберем $x = min(a_{i+1}, a_{i+2})$. Заменим a_i и a_{i+3} на $a_i - 1$, a_{i+1} и a_{i+2} на x.
3. Не может быть плохого интервала длины 4 в начале или конце пика (кроме случая, когда он занимает весь пик). Для этого выберем $x = min(a_{i+1}, a_{i+2})$. Заменим a_{i+3} на $a_i - 1$, a_i, a_{i+1}, a_{i+2} на x.
4. Если весь пик $[\ell, r]$ является плохим (все $b_i \neq a_i$), сравним количество индексов с $a_i > b_i$ и с $a_i < b_i$.
   - Если $a_i > b_i$ встречается не реже, поднимем пик до минимального $a_j > b_j$. Это не увеличит стоимость, но добавит совпадение $a_j = b_j$, что противоречит выбору решения с максимальным числом совпадений.
   - Иначе $a_i < b_i$ строго чаще. Тогда опустим весь пик на 1 — стоимость уменьшится, что противоречит оптимальности. Опускание невозможно только если соседи уже на 1 ниже: $b_{\ell-1} = b_\ell - 1$ или $b_{r+1} = b_r - 1$.

   В таком случае пик обязан быть длины 1 ($\ell = r$), причём $a_\ell < b_\ell$ и хотя бы один сосед находится на 1 ниже. Иначе можно опустить b_ℓ отдельно и улучшить стоимость.
5. Аналогично, плохой интервал внутри впадины имеет длину не более 3, а плохая впадина длины 1 удовлетворяет $a_\ell > b_\ell$ и $b_{\ell-1} = b_\ell + 1$ или $b_{\ell+1} = b_\ell + 1$.
6. Рассмотрим два подряд идущих одноэлементных плато: плохой пик $\{\ell\}$ и следующая за ним плохая впадина $\{\ell+1\}$ такие, что $b_{\ell+1} = b_\ell - 1$. Возможны два случая:
   - Если $a_\ell > b_{\ell-1}$ или $\ell = 1$: Тогда можно установить $b_\ell = a_\ell$ и $b_{\ell+1} = a_\ell - 1$. Стоимость не изменится, но появится новое совпадение $a_\ell = b_\ell$, что противоречит максимуму совпадений.
   - Если $a_\ell \leqslant b_{\ell-1}$: Тогда можно опустить оба значения до уровня соседа: $b_\ell = b_{\ell-1}$, $b_{\ell+1} = b_{\ell-1} - 1$. Это строго уменьшит стоимость, так как b_ℓ приблизится к a_ℓ (для плохого пика $a_\ell < b_\ell$), что противоречит оптимальности.

   Аналогично доказывается невозможность пары плохая впадина $\{\ell\} \to$ плохой пик $\{\ell+1\}$ с $b_{\ell+1} = b_\ell + 1$.
7. Следовательно, плохой пик не может быть окружён двумя плохими впадинами, и наоборот.

Таким образом, для любого b_i найдётся a_j ($|j - i| \leqslant 4$), такое что $b_i \in \{a_j - 1, a_j, a_j + 1\}$.