

Нетрадиционная игра

Автор задачи и разработчик: Павел Скобелкин

Группы 1 и 2

Для решения первой подзадачи можно было воспользоваться полным рекурсивным перебором случаев, в котором оба игрока будут выбирать для себя лучшие исходы соответственно. Оптимальная стратегия означает, что для каждого игрока надо было перебрать ходы противника, вернуть наиболее оптимальные достижимые противником состояния, и выбрать из всех своих ходов тот, который приводит к своему оптимуму среди рекурсивных результатов.

Для решения второй подзадачи достаточно было добавить меморизацию состояний перебора, заметив, что они часто будут повторяться.

Группы 7, 8, 9

В этих группах тестов было ограничение $k = 1$. Поймем, что будет происходить в таком случае.

Игра будет состоять из одного хода первого игрока, причем он хочет максимизировать сумму массива. В этой и последующих подзадачах обозначим $d_i = (a_i \oplus M) - a_i$ (где $M = 2^x - 1$) — выгода, которую игрок получит, если «возьмет» этот элемент. Финальный счет игры будет равен изначальной сумме массива a плюс сумме на отрезке в массиве выгод. Сумма массива a — константа, получается, в этой подзадаче необходимо было найти подотрезок массива d с максимальной суммой.

В зависимости от эффективности, можно было получить разный балл: седьмую подзадачу проходило решение с асимптотикой $\mathcal{O}(n^3)$: переберем отрезок $[l, r]$ за $\mathcal{O}(n^2)$, посчитаем на нем сумму за $\mathcal{O}(n)$ и возьмем максимум по всем отрезкам.

Решение седьмой подзадачи можно оптимизировать до $\mathcal{O}(n^2)$, используя префиксные суммы, и тем самым пройти **восьмую** подзадачу.

Девятую подзадачу можно было пройти, также используя префиксные суммы: $d_l + d_{l+1} + \dots + d_r = \text{pref}_{r+1} - \text{pref}_l$. Таким образом, для максимизации суммы на отрезке, если правая граница фиксирована, нужно выбирать левую границу с минимальной префиксной суммой. Это возможно сделать за $\mathcal{O}(n)$, храня минимумы на префиксных суммах или воспользовавшись любым другим стандартным подходом поиска отрезка максимальной суммы.

Группы 3, 4

Для решения третьей подзадачи можно было воспользоваться полным перебором ходов.

Случай $k = 1$ уже разобран выше; при $k = 2$ для каждого хода первого игрока $[l_1, r_1]$ (таких ходов $\mathcal{O}(n^2)$) найдем оптимальный ход второго игрока. Возможных ходов второго игрока также $\mathcal{O}(n^2)$, после этого счет игры можно посчитать за $\mathcal{O}(n)$. Таким образом, это решение работает за $\mathcal{O}(n^5)$.

Используя технику префиксных сумм, можно оптимизировать предыдущее решение (чтобы не считать счет игры за $\mathcal{O}(n)$). Таким образом получить асимптотику $\mathcal{O}(n^4)$ и пройти четвертую подзадачу.

Группа 5

Далее рассмотрим решение пятой подзадачи за $\mathcal{O}(n^3)$. Также переберем ход первого игрока, и для каждого хода первого игрока найдем самый оптимальный ход второго, но будем искать его более эффективно.

Пусть первый игрок сделал ход $[l, r]$. Тогда есть несколько вариантов того, куда второй игрок будет делать ход:

1. в $[0, l)$, то есть строго слева от первого отрезка;
2. в $(r, n - 1]$, то есть строго справа;

3. в $[l, r]$, то есть внутри первого хода;
4. пересекая ровно одну из двух границ;
5. в отрезок, полностью содержащий $[l, r]$.

Теперь, для пяти возможных вариантов, нужно решить задачу за $\mathcal{O}(n)$. Несложно понять, что первые три варианта аналогичны случаю $k = 1$ на подотрезках массива: решается за линейное время тем же способом. Для вариантов 4 и 5 нужно посчитать оптимальные ходы вида $[i, l]$, $[l, j]$, $[i, r]$ и $[r, j]$, что можно сделать за $\mathcal{O}(n)$ суммарно, просто перебирая i и j по порядку.

Группа 6

Для решения шестой подзадачи можно было заметить, что все вышеописанные значения можно было пересчитывать быстрее.

Давайте зафиксируем левую границу первого хода и будем перебирать правую границу по возрастанию. Научимся пересчитывать минимум во всех пяти случаях.

Случай 3 обрабатывается за $\mathcal{O}(n)$ для фиксированного l и всех r аналогично случаю нечётного k (ищем максимальный по сумме подотрезок на $[l, r]$, пересчитывается за $\mathcal{O}(1)$ с каждым увеличением r на единицу).

Для случаев 1 и 2 нужно научиться считать отрезок с минимальной суммой на префиксе и суффиксе массива d . Это предподсчитывается за $\mathcal{O}(n)$ одним проходом с начала и одним проходом с конца.

В случае 5 начало отрезка l' лежит до l , а конец r' — после r . Таким образом, его сумма имеет вид $\min(d'_{l'} - d'_{r'})$, где $l' \leq l$ и $r' \geq r$. Чтобы находить максимум, достаточно знать минимум и максимум на префиксе и суффиксе, что тоже предподсчитывается. При этом $d'_{l'} = d_{l'}$, а $d'_{r'} = d_{r'} + \Delta$, где Δ — изменение суммы массива b после хода первого игрока.

Случай 4, когда пересекается только одна из двух границ, обрабатывается похоже — например, слева берется все еще максимум, а справа — минимум d' на $[l, r]$, что можно считать в процессе при движении границы r .

Итого решение за $\mathcal{O}(n^2)$.

Группа 10

Для десятой подзадачи заметим, что все элементы массива равны, а значит все значения d_i также равны одному и тому же числу $c = 2^x - 1 - 2a_1$. Так как $2^x - 1$ нечетно, число c не может быть равно нулю.

Если $c > 0$, то переворачивать любой элемент выгодно. Тогда при нечетном k первый игрок просто выбирает весь массив и получает прибавку nc . При четном k второй игрок может повторить ход первого и вернуть массив в исходное состояние, поэтому итоговая прибавка будет равна 0.

Если $c < 0$, то каждый элемент, перевернутый нечетное число раз, только ухудшает ответ для первого игрока. При нечетном k ему выгодно выбрать отрезок длины 1, поэтому ответ равен исходной сумме плюс c .

Разберем случай четного k . Пусть первый игрок выбрал отрезок $[l, r]$. Тогда второй игрок хочет максимизировать количество элементов, перевернутых нечетное число раз.

Он может выбрать, например:

- отрезок $[1, l - 1]$, тогда нечетно перевернутыми останутся r элементов;
- отрезок $[r + 1, n]$, тогда нечетно перевернутыми останутся $n - l + 1$ элементов;
- отрезок $[1, n]$, тогда нечетно перевернутыми останутся все элементы вне $[l, r]$, то есть $n - (r - l + 1)$ элементов.

Следовательно, после выбора первого игрока $[l, r]$ второй может гарантировать не менее чем

$$\max(r, n - l + 1, n - (r - l + 1))$$

нечетно перевернутых элементов. С другой стороны, несложный разбор всех взаимных положений двух отрезков показывает, что больше добиться нельзя.

Значит итог после выбора первого игрока $[l, r]$ равен

$$\text{base} + c \cdot \max(r, n - l + 1, n - (r - l + 1)).$$

Остается минимизировать это выражение по всем l и r . Обозначим

$$a = l - 1, \quad b = n - r, \quad p = r - l + 1.$$

Тогда $a + b + p = n$, а

$$\max(r, n - l + 1, n - (r - l + 1)) = \max(a + p, p + b, a + b) = n - \min(a, b, p).$$

Следовательно, первому игроку нужно максимизировать $\min(a, b, p)$, то есть разбить массив на три части как можно более равномерно. Поэтому

$$\min_{1 \leq l \leq r \leq n} \max(r, n - l + 1, n - (r - l + 1)) = n - \left\lfloor \frac{n}{3} \right\rfloor = \left\lceil \frac{2n}{3} \right\rceil.$$

Итак, для десятой подзадачи ответ равен

$$\text{base} + \begin{cases} nc, & \text{если } k \text{ нечетно и } c > 0, \\ 0, & \text{если } k \text{ четно и } c > 0, \\ c, & \text{если } k \text{ нечетно и } c < 0, \\ c \cdot \left\lceil \frac{2n}{3} \right\rceil, & \text{если } k \text{ четно и } c < 0. \end{cases}$$

Группа 12

В этой подзадаче массив d будет иметь вид $d_1 = c, \quad d_2 = -c, \quad d_3 = c, \quad d_4 = -c, \dots$. Значит сумма на любом отрезке может быть только одной из трех: $c, 0, -c$.

Рассмотрим случай четного k . Так как сумма любого отрезка может быть только равна $c, 0$ или $-c$, а итог после двух ходов равен сумме на симметрической разности двух отрезков, то возможных значений тоже константное число. После перебора всех случаев по четности границ получается следующая формула.

Если $n \leq 2$, то второй игрок всегда может полностью нейтрализовать ход первого, поэтому ответ равен

$$\text{base}.$$

Если $n \geq 3$ нечетно, то ответ равен

$$\text{base} + \begin{cases} 0, & \text{если } c > 0, \\ 2c, & \text{если } c < 0. \end{cases}$$

Если $n \geq 4$ четно, то ответ равен

$$\text{base} - |c|.$$

Группа 14

В этой подзадаче для каждого i выполнено $a_i < 2^{x-1}$. Тогда

$$d_i = (2^x - 1) - 2a_i \geq 1,$$

то есть все значения d_i строго положительны. Поэтому в случае нечетного k первому игроку выгодно перевернуть весь массив, и ответ будет равен

$$\sum_{i=1}^n ((2^x - 1) - a_i) = n(2^x - 1) - \sum_{i=1}^n a_i.$$

При четном k второй игрок может повторить ход первого и вернуть массив в исходное состояние. Поскольку все d_i положительны, никакой ход второго игрока не может сделать итог меньше исходной суммы, значит ответом будет просто исходная сумма массива.

Наблюдение

Здесь же сделаем **важное наблюдение**: случай нечетного k эквивалентен $k = 1$. Аналогично, случай произвольного четного k эквивалентен $k = 2$ по тем же самым причинам. Докажем это свойство формально:

- Пусть k нечетное. Пусть результат игры за один ход это A . Тогда первый может гарантировать себе результат A : делает первый ход, максимизируя сумму, а далее повторяет (отменяет) все ходы соперника. Покажем, что второй тоже может гарантировать себе результат A : он будет просто повторять все ходы первого, тем самым оставив первому один ход в самом конце.
- Пусть k четное. Пусть первый и второй сделали ход — тогда далее первый может повторять ходы второго, тем самым не изменив результат после двух ходов. Аналогично, второй может повторять все ходы первого, кроме последнего хода, тем самым гарантировав себе результат за два хода.

Таким образом, решения групп 3 – 6 также являются решениями групп 15 – 17.

Группа 13

В тринадцатой подзадаче подразумевалось сведение массива большой длины к массиву меньшей длины. Сожмем блоки одинаковых чисел, после будем рассматривать каждое действие как затрагивающее два блока частично и все блоки посередине целиком. Для блоков, затронутых целиком, можно просто хранить и использовать общее d_i . Для крайних блоков число затронутых элементов аккуратно перебирается или оптимально определяется с учетом логики, описанной выше.

Полное решение

Заметим, что случай $k = 1$ уже разобран выше и тривиален: все сводится к поиску отрезка максимальной суммы. Далее будем рассматривать только случай $k = 2$ и эффективно искать ход первого игрока, при котором оптимальный ход второго игрока наиболее выгоден первому.

Пусть первый игрок выбрал отрезок $[l, r]$, а второй — отрезок $[l', r']$. Тогда итоговая прибавка к сумме массива равна сумме d_i по симметрической разности этих двух отрезков. Обозначим через $f(l, r)$ минимальную прибавку, которую сможет оставить второй игрок после хода первого игрока $[l, r]$. Тогда требуется найти $\max_{1 \leq l \leq r \leq n} f(l, r)$.

Будем делать бинарный поиск по ответу. То есть проверять предикат: можно ли первому игроку гарантировать прибавку не меньше, чем X (монотонный предикат).

Итак, остается научиться быстро проверять, существует ли отрезок $[l, r]$, для которого

$$f(l, r) \geq X.$$

Сделаем это с помощью «разделяй и властвуй». Пусть функция $\text{check}(L, R, X)$ проверяет, существует ли подходящий отрезок $[l, r]$, полностью лежащий в $[L, R]$. Если $L = R$, то отрезок только один, и его значение можно посчитать напрямую за $\mathcal{O}(1)$, используя заранее посчитанные минимумы на префиксах и суффиксах.

Иначе положим

$$M = \left\lfloor \frac{L + R}{2} \right\rfloor.$$

Тогда искомый отрезок либо:

1. целиком лежит в $[L, M]$;
2. целиком лежит в $[M + 1, R]$;
3. пересекает середину, то есть имеет вид $[l, r]$, где $l \leq M < r$.

Первые два случая обрабатываются рекурсивно. Остается рассмотреть только отрезки, пересекающие середину.

Для каждой левой части $[l, M]$ построим тройку чисел (A_l, B_l, C_l) :

- A_l — вклад левой части, если второй игрок выбирает отрезок целиком справа от середины;
- B_l — минимальный вклад левой части, если отрезок второго игрока пересекает середину;
- C_l — минимальный вклад левой части, если отрезок второго игрока целиком лежит слева от середины.

Аналогично, для каждой правой части $[M + 1, r]$ построим тройку (A_r, B_r, C_r) :

- A_r — минимальный вклад правой части, если второй игрок целиком справа от середины;
- B_r — минимальный вклад правой части, если второй игрок пересекает середину;
- C_r — вклад правой части, если второй игрок целиком слева от середины.

Тогда для отрезка первого игрока $[l, r]$ второй игрок имеет ровно три типа оптимальных ответов относительно середины:

1. его отрезок целиком слева от середины;
2. его отрезок пересекает середину;
3. его отрезок целиком справа от середины.

Следовательно, первый игрок может гарантировать значение не меньше X тогда и только тогда, когда одновременно выполнены три неравенства

$$A_l + A_r \geq X, \quad B_l + B_r \geq X, \quad C_l + C_r \geq X.$$

Значения (A_l, B_l, C_l) для всех $l = M, M - 1, \dots, L$ и значения (A_r, B_r, C_r) для всех $r = M + 1, M + 2, \dots, R$ можно посчитать за линейное время, если двигать границы по одному и поддерживать:

- сумму текущего отрезка;
- максимум суммы подотрезка внутри него;
- минимальный префикс;
- минимальный суффикс;
- а также заранее посчитанные минимумы на префиксах и суффиксах всего массива.

Теперь задача для пересекающих середину отрезков свелась к следующей: даны все левые тройки (A_l, B_l, C_l) и все правые тройки (A_r, B_r, C_r) ; нужно проверить, существует ли пара, для которой

$$A_l \geq X - A_r, \quad B_l \geq X - B_r, \quad C_l \geq X - C_r.$$

Это стандартная задача о доминировании в трех измерениях.

Решим ее так:

1. отсортируем левые тройки по первой координате A по убыванию;
2. для каждой правой тройки создадим запрос с порогами

$$(X - A_r, X - B_r, X - C_r);$$

3. отсортируем запросы по первой координате по убыванию;

4. будем sweep-line'ом добавлять в структуру все левые тройки, у которых первая координата уже достаточна;
5. по второй координате сделаем сжатие координат, а в Fenwick tree будем хранить максимум третьей координаты среди уже добавленных троек.

Тогда для каждого запроса можно за $\mathcal{O}(\log n)$ проверить, существует ли добавленная левая тройка, у которой одновременно

$$B_l \geq X - B_r \quad \text{и} \quad C_l \geq X - C_r.$$

Значит все пересекающие середину отрезки проверяются за

$$\mathcal{O}((R - L + 1) \log(R - L + 1)).$$

Итак, одна проверка $\text{check}(L, R, X)$ работает за

$$\mathcal{O}(n \log^2 n) :$$

на каждом уровне рекурсии суммарный размер всех отрезков равен $\mathcal{O}(n)$, а работа на одном узле стоит $\mathcal{O}(\text{len} \log \text{len})$.

Остается сделать бинарный поиск по ответу. Прибавка для случая двух ходов не может быть больше 0, потому что второй игрок всегда может повторить ход первого. С другой стороны, она не меньше, чем

$$-\sum_{i=1}^n |d_i|.$$

Значит бинарный поиск идет по отрезку

$$\left[-\sum |d_i|, 0 \right].$$

Итоговая асимптотика: $\mathcal{O}(n \log^2 n \log C)$, где $C = \sum |d_i|$.