Palindromization

Автор задачи и разработчик: Владислав Власов

Для начала воспользуемся некоторыми преобразованиями. Они нужны для решения не всех подзадач, однако, сделав их, затем будет проще думать о том, что в задаче требуется, и почему время работы, например, полного перебора, будет достаточным, чтобы уложиться в ограничения.

Итак, рассмотрим наш массив a.

1. Заметим, что нет смысла делать прибавления на отрезках, пересекающих середину массива. Действительно, если m — середина массива, то добавление x на отрезке $[m-p, m+q]$, где $p < q$, с точки зрения цели задачи равносильно добавлению x на отрезке $[m+p+1, m+q]$. Действительно, разница между этими двумя действиями только в прибавлении x на $[m-p, m+p]$, а это действие не изменяет свойство a быть палиндромом.

2. Теперь сведем задачу к другой: рассмотрим массив b вида

$$b = [a_1 - a_n, a_2 - a_{n-1}, \ldots, a_{\lfloor \frac{n}{2} \rfloor} - a_{\lceil \frac{n+1}{2} \rceil + 1}],$$

   то есть массив разностей противоположных элементов массива a. Если a — палиндром, то все элементы b — нули. А прибавление x на отрезке массива a равносильно прибавлению или вычитанию x на отрезке массива b.

3. Теперь посчитаем c — разностный массив массива b, к которому дописали 0 в начало и в конец, то есть

$$c = [b_1, b_2 - b_1, b_3 - b_2, \ldots, b_{\lfloor \frac{n}{2} \rfloor} - b_{\lfloor \frac{n}{2} \rfloor - 1}, 0 - b_{\lfloor \frac{n}{2} \rfloor}].$$

   В таком массиве почти ничего не меняется в плане нашей цели — когда все элементы b равны нулю, все элементы c равны нулю, и наоборот. Однако теперь мы свели прибавление x на отрезке b к прибавлению x к одному элементу c и вычитанию x из другого.

Подзадача 1

В первой подзадаче можно было написать аккуратный полный перебор возможных действий. Будем анализировать массив b вместо массива a. В нем числа лежат от -9 до 9, и за каждое действие достаточно выбрать два числа одного знака и прибавить или вычесть некоторое значение на отрезке между ними.

Заметим, что порядок действий не важен, а значит можно идти по массиву b от начала к концу и для каждого очередного b_i выбирать ему конец отрезка, элемент на котором того же знака, и делать прибавление на отрезке. Массив b имеет размер не более 5, поэтому различных отрезков для прибавления не больше $\frac{5 \cdot 4}{2} = 10$ (на самом деле еще меньше, если концы отрезков выбирать одного знака).

При чем видно, что эвристически делать прибавления меньших x выгодно только тогда, когда в b исходного много чисел $< k$, а в таком случае ответ будет меньше. Таким образом, у нас есть меньше 10 возможных отрезков для прибавления, на каждом есть несколько способов выбрать прибавляемое значение, и в конечном итоге полный перебор действий на тестах первой группы будет укладываться в ограничения. Единственное необходимое для этого наблюдение — что можно не рассматривать в a отрезки, пересекающие середину массива.

Подзадача 2

В этой подзадаче к каждому элементу нужно прибавить не больше чем 1. Снова будем рассматривать массив b, и заметим, что именно b_i операций нужно применить к i-му элементу. Можно доказать, что в таком случае ответом будет являться количество непрерывных отрезков из единиц и минус единиц в массиве b. Действительно, за каждое действие не получится уменьшить количество таких отрезков хотя бы на 2, значит ответ не может быть меньше.

Подзадача 3

Заметим следующий факт: если $a_i \leqslant a_{i+1}$ для всех i, то в массиве b все числа будут отрицательными, при чем неубывающими. В таком случае количество операций, применяемых к b_i, будет не меньше, чем количество операций, применяемых к b_{i+1}. Соответственно, применим следующий алгоритм:

1. Будем прибавлять k ко всем числам массива b, пока очередное прибавление не сделает его последний элемент положительным. Тогда вместо этого прибавим минус последний элемент b ко всем его элементам.

2. Мы добились того, что $b_{\text{last}} = 0$, за минимальное число действий. Теперь повторим те же действия, но уже для отрезка с первого элемента b до предпоследнего.

3. Когда последние два элемента b равны нулю, продолжим сужать префикс, на котором мы совершаем прибавления, постепенно придя к тому, что весь массив b состоит из нулей.

Факт, что мы использовали минимальное число действий, можно доказать конструктивно. Для этого достаточно показать, что любой способ, включающий в себя прибавление на двух непересекающихся отрезках, можно преобразовать в не худший способ, в котором этих отрезков нет. Это решается небольшим разбором случаев, который мы не будем здесь приводить, потому что это не является ключевой идеей, необходимой для решения.

Подзадача 4

Начиная с этой подзадачи, будем пользоваться массивом c для решения. Когда мы посчитали массив c, нашей задачей становится за минимальное число действий вида «прибавить x к одному элементу и вычесть из другого» получить полностью состоящий из нулей массив. В подзадаче с $k = 1$ это достигается достаточно прямолинейным образом: нужно за каждую операцию прибавлять 1 к отрицательному элементу и вычитать 1 из положительного.

Поскольку сумма элементов массива c равна нулю, то можно необходимое количество действий просто посчитать как сумму всех его положительных элементов. Время работы решения — $\mathcal{O}(n)$.

Подзадача 5

Аналогично с прошлой подзадачей, построим массив c, но обработаем немного по-другому. Теперь мы можем прибавлять и вычитать 1 или 2. На число c_i необходимо $\left\lceil \frac{|c_i|}{2} \right\rceil$ операций, меньше невозможно. Посчитаем две суммы таких величин: для положительных значений и для отрицательных. Ответом будет являться большая из них.

Для этого сначала заметим, что нет смысла вычитать из отрицательных чисел и прибавлять к положительным — тогда всегда можно либо уменьшить ответ, либо избавиться от такой операции, не увеличив его. А затем заметим, что меньшего количества операций добиться нельзя, а за такое количество операций получить все нули можно. В той части, в которой получилась максимальная сумма $\left\lceil \frac{|c_i|}{2} \right\rceil$, каждое c_i обработаем как $\left\lfloor \frac{|c_i|}{2} \right\rfloor$ операций прибавления или вычитания двойки, и еще возможно одну операцию прибавления или вычитания единицы. В другой части сумма получилась меньше, значит в ней было меньше нечетных чисел. Тогда каждому нечетному все еще хватит по единице, а «лишние» единицы сгруппируем по две, чтобы добавить к тем четным, которым не хватило двоек.

Подзадача 6

Для последней подзадачи необходимо было сделать еще несколько конструктивных наблюдений. Для начала переформулируем задачу как «дано множество положительных c_i и множество отрицательных c_i, требуется найти такое множество слагаемых от 1 до 3, что на них можно разложить все числа как первого, так и второго множества». Действительно, каждая операция — это прибавление к отрицательному и вычитание из положительного. По сути мы раскладываем на слагаемые множества положительных и отрицательных чисел среди c_i.

Теперь заметим, что:

- Любое разложение на слагаемые от 1 до 3 числа $c_i \geqslant 8$ всегда содержит в себе несколько слагаемых, дающих в сумме ровно 6 (доказывается аккуратным разбором случаев). Из этого следует, что вместо разложения $c_i \geqslant 8$ можно независимо раскладывать $c_i - 6$ и 6.

- В разложении чисел $c_i = 5$ или $c_i = 7$ всегда найдется несколько слагаемых с суммой 3. Значит можно раскладывать независимо $c_i - 3$ и 3.

Тогда разделим c на группу отрицательных и группу положительных чисел и преобразуем описанным образом. В каждой группе останутся только числа из множества $\{1, 2, 3, 4, 6\}$. Теперь приведем решение, позволяющее найти ответ за время $\mathcal{O}(n^2)$. Для этого заметим, что количество чисел в каждом таком наборе, не равных 6, линейно от n (из каждого исходного c_i получилось не более двух чисел от 1 до 4). Более того, не может быть так, чтобы в обоих наборах было разложение $6 = 2+2+2$, так как тогда его можно заменить на $6 = 3 + 3$ с меньшим числом слагаемых. Аналогичное верно для единиц.

Тогда хотя бы в одном наборе количество единиц не превосходит $\mathtt{cnt}[1] + 2\mathtt{cnt}[2] + 3\mathtt{cnt}[3] + 4\mathtt{cnt}[4] + 6$, а количество двоек — $\mathtt{cnt}[2] + \mathtt{cnt}[3] + 2\mathtt{cnt}[4] + 3$. На самом деле будет видно, что необходимое количество единиц и двоек еще меньше, но пока нам хватит факта, что они линейны от n.

Переберем тогда количество используемых единиц (m_1) и двоек (m_2), найдем количество троек как $\frac{\mathtt{total} - m_1 - 2m_2}{3}$, и проверим, можно ли оба множества разбить на такие слагаемые. Чтобы проверить, можно ли разбить множество на такие слагаемые, достаточно проверить следующие условия:

1. Если $m_1 < \mathtt{cnt}[1]$, то нельзя, так как единицы получить по-другому нельзя. Иначе — уменьшим m_1 на $\mathtt{cnt}[1]$ и забудем про них пока.

2. Если $m_2 \leqslant \mathtt{cnt}[2]$, то на оставшиеся двойки и все четверки должно хватить единиц, все остальное разложится на слагаемые, равные 3.

3. Если $m_2 \leqslant \mathtt{cnt}[2] + 2\mathtt{cnt}[4]$, то на оставшиеся четверки должно хватить единиц. Более того, если $m_2 - \mathtt{cnt}[2]$ нечетно, то оставшаяся двойка явно потребует лишней единицы. Действительно, либо в разложении какого-то числа придется поставить $2+1$, либо из текущего распределения двоек придется одно разложение убрать, и в нем использовать единицы вместо двоек. Небольшой разбор случаев того, как слагаемые 2 можно распределить по «целям» показывает, что одной лишней единицы хватает, чтобы оставшиеся числа распадались на слагаемые 3, а без нее разложить не получится.

4. Если $m_2 \leqslant \mathtt{cnt}[2] + 2\mathtt{cnt}[4] + 3\mathtt{cnt}[6]$, то разобьем все как $2 = 2$, $4 = 2 + 2$ и $6 = 2 + 2 + 2$, и останется либо одна, либо две лишние двойки. Еще один небольшой разбор случаев показывает, что на каждую из оставшихся лишних двоек понадобится лишняя единица вне зависимости от того, распределять двойки показанным образом или нет.

5. Если $m_2 > \mathtt{cnt}[2] + 2\mathtt{cnt}[4] + 3\mathtt{cnt}[6]$, то аналогично, на каждую лишнюю двойку понадобится по единице, чтобы объединить их в $3 = 1 + 2$.

6. Если $m_2 > \mathtt{cnt}[2] + 2\mathtt{cnt}[4] + 3\mathtt{cnt}[6] + \mathtt{cnt}[3]$, то не хватит «мест», в которые двойки можно распределить.

Таким образом, для каждой пары (m_1, m_2) можно определить, дают ли они корректное разбиение, и выбрать ту, которая минимизирует количество слагаемых в целом. Есть также альтернативное решение, сводящее эту задачу к рюкзаку, однако мы здесь его приводить не будем.