Разбор задачи «Плохая погода»

Все, что требовалось сделать в задаче — сравнить время, которое Кеша затратит на одну и другую дороги. Чтобы не было проблем с точностью, можно было провести вычисления в целых числах. Если же вычисления выполнялись в вещественных числах, следовало проводить сравнение двух чисел с точностью до некоторой небольшой величины: если разность между посчитанными временами оказывалась меньше этой величины, их следовало считать равными.

Разбор задачи «Скоростной трамвай»

Как понятно из условия, скоростной трамвай может обогнать ровно один обычный трамвай на кольце. Разумеется, если он его к этому моменту догонит. Можно перебирать кольца, начиная с первого, можно перебирать обычные трамваи, начиная с последнего (того, который отправится в путь непосредственно перед скоростным) — в любом случае нам нужно будет выяснить, когда скоростной трамвай догонит обычный.

Если мы перебираем кольца, то для каждого кольца подсчитаем время прибытия обычного трамвая, непосредственно предшествующего скоростному, и скоростного. Если окажется, что скоростной мог бы доехать до кольца быстрее, то это значит, что на самом деле он прибудет на кольцо в тот же момент, что и обычный, после чего обгонит его. Тогда мы переходим к следующему кольцу и следующему обычному трамваю. Если же окажется, что скоростной трамвай не догонит обычный (заметим, что здесь нужно, как и в задаче A, аккуратно проверять случай равенства), то мы переходим к следующему кольцу, а обычный трамвай остается прежним.

Удобно из соображений общности считать, что в конце маршрута располагается еще одно кольцо, и обрабатывать его так же, как и все «реальные» кольца.

Разбор задачи «Правильный путь»

Внимательный анализ условия и примера приводит к выводу, что пассажир p_j поменяется местами со всеми пассажирами, которые стоят ближе него к выходу, и при этом для каждого из них $p_i > p_j$. Чтобы получить ответ, посчитаем для каждого пассажира p_j количество таких пассажиров ближе него к выходу (соответствующий математический термин — количество инверсий).

Ограничения на количество пассажиров позволяют выполнить такой подсчет непосредственно.

Разбор задачи «Трамвайная остановка»

Разделим ожидающих трамвая на три группы. В первую поместим всех, кого уже обрызгали, во вторую - тех из остальных, кто был на остановке изначально, и в третью - всех остальных, то есть тех, кто пришел за время событий, описанных в задаче. Для простоты реализации пополним первую группу каким-нибудь участником, который в начальный момент времени стоит ближе всего к дороге.

Заметим, что любой представитель первой группа находится на расстоянии s_{min} от дороги. Действительно, в начальный момент времени это выполняется по условию распределения на группы, проезжающая машина не изменяет этого свойства, а любой пришедший потенциальный пассажир не меняет значение s_{min}.

Кроме того, заметим, что все представители третьей группы приходят в порядке неубывания. Действительно, s_{min} и s_{max} только увеличиваются после каждой обработки сообщения.

Первую группу будем хранить парой чисел - количеством представителей и их (общим) положением. Упорядочим всех ожидающих в начальный момент времени по неубыванию расстояния от дороги и поместим в двустороннюю очередь (англ. deque). Аналогично, при помощи двусторонней очереди будем обрабатывать всех приходящих. Поскольку вторая группа не пополняется, а третья пополняется в порядке неубывания, обе двусторонние очереди все время будут упорядочены.

Теперь будем обрабатывать сообщения.

Если к нам приходит сообщение о машине, вытащим с начала очередей второй и третьей группы всех обрызганных ею пассажиров и переместим в группу обрызганных (не модифицируя ее положение). После этого проверим, обрызгала ли машина первую группу. Если первая группа не была обрызгана, то никто не был (они стоят ближе всех). Если первая группа была обрызгана, то ее

размер - количество обрызганных пассажиров. После этого переместим всю первую группу, если, конечно, ее положение изменилось.

Если к нам приходит новый пассажир, то положение первой группы есть s_{min}, положение стоящего дальше всех пассажира из первой, второй или третьей — s_{max}(в последних случаях нужно посмотреть на последний элемент очереди, если он, конечно, есть). Теперь мы можем вычислить $(s_{min} + s_{max})/2$ и поместить пассажира в группу номер три.

Хотя каждый запрос и может в худшем случае выполняться за $O(n)$, итоговая сложность решения при использовании сортировки за $O(n \cdot \log n)$ будет $O(n \cdot \log n + q)$. Доказательство этого факта оставим читателю в качестве упражнения.

Разбор задачи «Парк»

Поскольку задача очень сложная и детальный разбор её займет много страниц текста, опишем только основные утверждения и не будем вдаваться в точные доказательства и детали реализации решения.

Будем называть направление влево-вправо горизонталью, направление вверх-вниз вертикалью. Не нарушая общности, будем далее считать, что первый шаг делается вправо, а первый шаг по вертикали делается вверх. Действительно, мы всегда можем просто перенаправить оси.

Утверждение 1. Всегда существует ответ, в котором ровно один шаг вправо и ровно один шаг вниз. Искомый периметр есть дважды сумма всех чисел ввода, кроме первого и того, которое выбрано как шаг вверх.

Действительно, вторая часть утверждения накладывает ограничение сверху на искомый периметр, которое может быть достигнуто, если делать вправо только первый шаг и делать вверх только первый шаг из тех, что обозначены как шаги по вертикали.

Изменим теперь постановку задачи. Нам нужно определить, какие из шагов будут по вертикали. Все остальные шаги будут по горизонтали.

Утверждение 2. Ограничение начального положения равносильно следующему: первый шаг по горизонтали меньше суммы остальных шагов по горизонтали и первый шаг по вертикали меньше суммы остальных шагов по вертикали.

Действительно, вопрос "лежит ли точка на границе прямоугольника"может быть решен раздельно для горизонтального и вертикального направления, и решение его выглядит именно так.

Из общей суммы и суммы вертикальных шагов мы выразить сумму горизонтальных шагов. Поскольку первый горизонтальный шаг мы знаем, утверждение 2 в части шагов по горизонтали может быть сформулировано как некоторое ограничение S на сумму шагов по вертикали сверху.

Задача теперь равносильна задаче выбора поднабора вертикальных шагов из некоторого набора (начальный, кроме первого элемента), со следующими условиями:

Условие 1 (о сумме). Сумма выбранного поднабора не больше S.

Условие 2 (о первом элементе). Первый элемент выбранного поднабора меньше суммы остальных.

Условие 3 (минимизации). Из всех поднаборов, для которых выполнены условия 1 и 2, надо выбрать поднабор с минимальным первым элементом.

Будем решать эту задачу, рассматривая числа набора справа налево и проверяя, может ли заданный элемент быть первым элементом искомого поднабора.

Пусть нам уже известен какой-либо поднабор, находящийся полностью правее текущего элемента. Если рассматриваемое число не меньше первого числа известного поднабора, то нам не нужно рассматривать это число, поскольку ответа лучше мы не получим. Если рассматриваемое число меньше первого числа поднабора, то рассматриваемое число в паре с первым числом известного поднабора будет удовлетворять условию 1 (о сумме) и условию 2 (о первом элементе) и мы нашли наилучший возможный ответ по условию 3 (минимизации). Остается рассмотреть случай, когда выполняется...

Условие 4 (ответ не найден). Ни один поднабор правее рассматриваемого элемента не подходит по условию 1 (о сумме) или 2 (о первом элементе).

Кроме того, из условий 1 и 2 однозначно следует, что первое число меньше $\frac{S}{2}$. Проверим это утверждение и будем считать, что выполняется...

Условие 5 (число не очень большое). Рассматриваемый элемент меньше $\frac{S}{2}$.

Выпишем теперь все числа, которые расположены не раньше рассматриваемого числа и которые в сумме с рассматриваемым не нарушают условие 1 (о сумме). Выписанное будем называть *преднабором*. Само первое число попадет в преднабор по условию 5 (число не очень большое). Собственно, искомый поднабор есть поднабор преднабора, поскольку числа, в паре с первым нарушающие условие 1 (о сумме), в искомый поднабор попасть никак не могут.

Пусть в преднаборе хотя бы одно число больше, чем первое. Тогда первое число, взятое с ним в паре, будет удовлетворять условию 1 (о сумме) и условию 2 (о первом элементе) и мы нашли наилучший возможный ответ по условию 3 (минимизации). Остается рассмотреть случай, когда выполняется...

Условие 6 (первое число самое большое). Первое число не меньше никакого числа преднабора.

Утверждение 3 (преднабор упорядочен). Пусть e и f — числа преднабора, и e расположено раньше f. Тогда $e \geqslant f$.

Действительно, если число e первое, то это обеспечивается условием 6. В противном случае из условий 6 и 5 получаем, что $e+f < S$, а это условие 1 (минимизации) для поднабора (e, f). Условие 4 (ответ не найден) требует, чтобы условие 2 (о первом элементе) для этого поднабора было нарушено, что и записано.

Теперь разобьем преднабор на левую и правую (префикс и суффикс) части $x_1, x_2, \ldots, x_m, y_1, y_2, \ldots, y_l$ по следующему условию:

Условие 7 (разбиения). Любое из y не меньше суммы всех последующих чисел. Для последнего из x это условие не выполняется. Математически: $y_j \geqslant \sum\limits_{i=j+1}^{l} y_i$ и $x_m < \sum\limits_{i=1}^{l} y_i$

Утверждение 4 (правая часть не пуста). $l \geqslant 1$.

Действительно, все числа положительны и условие 7 (разбиения) не может допустить обратного.

Если первое число попало в правую часть преднабора, то условие 2 (о первом элементе) не может быть выполнено. Остается рассмотреть случай, когда верно...

Условие 8 (левая часть не пуста). $m \geqslant 1$.

Утверждение 5 (не более двух левых). Пусть x_a, x_b и x_c — три числа левой части преднабора и $a < b < c$. Тогда верно, что $x_a + x_b + x_c > S$.

Действительно, поднабор $x_m, y_1, y_2, \ldots, y_l$ удовлетворяет по условию 7 (разбиения) условию 2 (о первом элементе). Кроме того, если $m = 1$, то таких троек просто нет, и утверждение верно. Значит, $m > 1$ и к этому поднабору применимо условие 4 (ответ не найден), и, стало быть, этот поднабор нарушает условие 2 (о сумме). Распишем, используя утверждение 3 (преднабор упорядочен) и условие 7 (разбиения): $x_a + x_b + x_c \geqslant x_m + y_1 + y_1 \geqslant x_m + y_1 + \sum\limits_{i=2}^{l} y_i = x_m + \sum\limits_{i=1}^{l} y_i > S$.

Утверждение 6 (выбора нет). Либо $m = 1$, либо x_2 входит в искомый поднабор. Числа $x_3, x_4, \ldots, x_m$ никогда не входят в искомый поднабор.

Действительно, если мы найдем некоторый ответ, не содержащий x_2, мы можем заменить в нем x_1 на x_2 и получить противоречие условию 4 (ответ не найден). Поскольку числа x_1 и x_2 обязательно присутствуют в искомом поднаборе, утверждение 5 (не более двух левых) не допускает после этого числа $x_3, x_4, \ldots, x_m$.

Таким образом, нам осталось научиться выбирать нужные числа из правой части $y_1, y_2, \ldots, y_l$. Нам достаточно найти наибольшее по сумме подмножество $y_1, y_2, \ldots, y_l$ такое, что сумма этого подмножества не больше $S - x_1 - x_2$ (если $m = 1$, то $S - x_1$) и проверить для получившегося поднабора условие 2 (о первом числе). Задача поиска такого подмножества решается жадно слева направо [например, см. разбор задачи D предыдущего этапа] в силу построения правой части преднабора.

Наивная реализация описанного выше подхода потребует $O\left(n^2\right)$ времени.

Утверждение 7 (правая часть быстро растет). $y_i \geqslant 2^{l-i-1} \cdot y_l$; $l = O\left(\log\left(\frac{\max a_i}{\min a_i}\right)\right)$, где a_i - набор расстояний из входного файла.

Действительно, эту оценку можно получить из условия 7 (разбиения) по индукции. Вторая часть очевидно следует из первой, если вспомнить, что y есть числа из a.

Утверждение 8 (преднабор почти не меняется). Любое число, меньшее $\frac{S}{2}$, попадет в преднабор для всех элементов левее. Если в преднабор попадет число, не меньшее $\frac{S}{2}$, то ответ будет найден перед проверкой условия 6 (первое число самое большое) на текущей итерации.

Действительно, мы строим преднабор для первых чисел, меньших $\frac{S}{2}$, а сумма чисел, меньших $\frac{S}{2}$, меньше S. Кроме того, число, меньшее $\frac{S}{2}$, будет меньше любого числа, не меньшего $\frac{S}{2}$.

Таким образом, при переходе от одного первого числа к другому при построении преднабора достаточно пополнить доставшийся от предыдущего построения преднабор, предварительно удостоверившись, что текущее первое число в паре с наименьшим из не меньших $\frac{S}{2}$ не образует ответ. Поскольку на каждой итерации мы рассматриваем только первые два числа левой части преднабора, итоговая сложность решения будет $O(n \cdot l) = O\left(n \cdot \log\left(\frac{\max a_i}{\min a_i}\right)\right)$.