Разбор задачи «Строгий Город»

Будем находить ответ для каждого запроса независимо. Нетрудно заметить, что для нахождения кратчайшего пути между заданными перекрестками нам достаточно рассмотреть лишь те дороги, которые являются ближайшими к одному из двух данных перекрестков в каждом направлении в каждой из сторон относительно перекрестка. А именно, если один из двух перекрестков запроса имеет координаты (X, Y), то нам достаточно рассмотреть ближайшую вертикальную дорогу с координатой не менее X, направленную вверх (в сторону увеличения координаты Y), ближайшую вертикальную дорогу с координатой не менее X, направленную вниз — две дороги "справа" от перекрестка. Так же нам могут потребоваться ближайшие дороги вверх и вниз, с координатой не более X — две дороги "слева" от перекрестка. Аналогично нам нужно добавить к рассмотрению по две ближайшие дороги "сверху" и "снизу" от перекрестка, направленные в разные стороны (если такие есть, разумеется). Аналогичным образом нужно добавить дороги для второго перекрестка запроса. Итого, у нас останется в рассмотрении максимум по 8 вертикальных и горизонтальных дорог, что в пересечении даст максимум 64 перекрестка, по которым следует искать путь. Оставив достаточно маленькое количество дорог и перекрестков можно построить граф (вершины — перекрестки, ребра — участки дорог между двумя соседними перекрестками) и найти кратчайшее расстояние, например, алгоритмом Дейкстры.

Разбор задачи «Выборы»

Заметим, что ответ равен -1 тогда и только тогда когда $2 \cdot \sum_{i=1}^{n}(a_i + b_i) \leqslant \sum_{i=1}^{n} k_i$ (поскольку мы можем фальсифицировать все доступные голоса). Иначе фальсификация возможна.

Пронумеруем проголосовавших на каждом участке. Не умаляя общности будем считать, что если на участке сделано y вбросов второго типа, то подменены голоса людей с номерами от 1 до y. Тогда для каждого человека мы можем посчитать цену его голоса — на сколько увеличится недовольство, если сфальсифицировать голос этого человека (в предположении, что голоса избирателей с меньшими номерами уже сфальсифицированы). Для пенсионеров она равна x, а для проголосовавшего избирателя с номером i на своем избирательном участке стоимость равна $i^2 - (i-1)^2 = 2i - 1$. Если мы хотим поменить голоса ценой не более Y, то на одном избирательном участке можно подменить не более $(Y + 1)/2$ голосов.

Рассмотрим множество людей, фальсификация голосов которых дает оптимальный ответ. Если существует человек, не принадлежащий этому множеству, цена голоса которого меньше цены голоса какого-то элемента этого множества, то мы можем улучшить ответ Следовательно, для любого человека в оптимальном множестве стоимость его голоса не превосходит стоимости голоса любого человека не входящего в оптимальное множество. Значит, существует такое число M, что стоимости голосов всех людей, входящих в оптимальное множество, не превосходят M, а стоимости голосов всех людей, не входящих в множество, больше или равны чем M. Искомое M можно найти двоичным поиском, внутри которого надо проверть, можно ли выиграть выборы с помощью фальсификации голосов людей с ценой голоса не больше M. Эта часть решения работает за $O(n \log \max(x, \max a_i))$.

Когда мы нашли M, посчитать ответ можно следующим образом: мы обязаны взять всех людей с ценой голоса строго меньше M, пусть их количество равно c. Тогда мы дополнительно должны сфальсифицировать $\lfloor (\sum_{i=1}^{n} k_i)/2 \rfloor - c + 1$ голос цены M. Посчитать ответ таким образом можно за $O(n)$, итого все решение работает за $O(n \log \max(x, \max a_i))$.

Разбор задачи «Львы»

Ключевое наблюдение — пусть Иван выбрал какое-то направление. Если есть лев, направление от Ивана на которого образует острый угол с выбранным направлением для движения, то лев может поймать Ивана, а иначе нет.

Отсюда есть несколько решений. Разберем наиболее простое. Рассмотрим выпуклую оболочку львов и Ивана. Если Иван внутри неё, то ответ NO, иначе YES, причем можно взять нормаль к стороне, на которой он лежит, ориентированную от многоугольника. Такое решение работает за $O(n \log n)$.

Также существует решение, не использующее выпуклую оболочку. Разобьем всех львов на два множества: в первом будут все львы, ордината которых больше, чем у Ивана, а также львы с

такой же ординатой, но большей абсциссой, а во втором — все остальные. Если одно из множеств пусто, то мы можем убежать, двигаясь вверх или вниз соответственно. Иначе заметим, что Иван не может бежать между львами из одного множества, так как в это случае направление одного из львов будет образовывать острый угол с направлением Ивана. Найдем в первом множестве львов, направление от Ивана к которым образует наименьший и наибольший угол и направлением налево (вектор $(1;0)$), пусть им соответствуют точки M_1 и N_1 соответственно. Найдем аналогичные точки M_2 и N_2 во втором множестве. Тогда Иван может бежать только между M_1 и M_2 или между N_1 и N_2. Обозначим точку, в которой находится Иван как O. Рассмотрим пару точек M_1 и M_2, вторая пара рассматривается аналогично. Если точки $M_1,\ O,\ M_2$ образуют поворот против часовой стрелки , то между этими точками Иван пробежать не может. Иначе он может бежать по нормали к стороне M_1O, ориентированной от треугольника M_1OM_2.

Это решение работает за $O(n)$.

Разбор задачи «Запасы воды»

Переформулируем задачу, используя понятия теории графов. Дано подвешенное дерево, в каждой вершине которого находится некоторый объем воды. Разрешено переливать любой объем воды из любой вершины в любого ее ребенка, а также доливать воду извне в любую вершину. Необходимо определить минимальный суммарный объем воды, который нужно долить извне, чтобы во всех вершинах было поровну воды.

В начале решения можно заметить тот факт, что можно ограничить доливание извне только доливанием в корень дерева. Действительно, доливание в любую другую вершину можно тогда выполнить в два шага: сначала долить необходимый объем воды в корень, затем по цепочке переливать этот объем в детей, пока он не окажется в нужной вершине. Из этого замечания сразу следует следующее решение. Заметим, что можно выполнить бинарный поиск по ответу: если существует решение, которое доливает суммарно x литров воды, то, очевидно, существует и решение, доливающее $y > x$ литров воды. Значит, остается лишь научиться проверять, можно ли равномерно распределить воду по вершинам, если долить в корень x литров воды.

Если мы знаем, что мы долили в корень x литров воды, то мы знаем, сколько воды в итоге должно будет оказаться в каждой вершине — это будет $(x + s)/n$, где s — суммарное количество воды в дереве изначально. После этого проверить возможность решения легко, если заранее вычислить размеры всех поддеревьев и объем уже имеющейся в них воды. Это можно сделать методом динамического программирования по поддеревьям. Затем, зная эту информацию, легко вычислить, сколько воды нужно долить в каждое из поддеревьев. Решение существует, если каждая из этих величин неотрицательна. Таким образом, мы можем за линейное время проверить, существует ли решение с данным значением x. Делая бинарный поиск по x, получим решение с асимптотикой $O(n \log(\varepsilon^{-1}))$, где ε — необходимая точность.

Более быстрое решение можно получить, если заметить, что условие «в каждое поддерево нужно долить неотрицательный объем воды» есть линейное неравенство на ответ. Действительно, пусть v — объем воды, который в итоге окажется в каждой вершине. Тогда в поддерево нужно долить неотрицательный объем воды, если объем уже имеющейся там воды меньше, чем размер поддерева, умноженный на v. Значит, минимально возможное значение v для этого поддерева равно отношению общего объема воды в поддереве к размеру поддерева. Общий же ответ для дерева равен максимальному значению этой величины по всем подддеревьям. Это решение работает за $O(n)$.

Разбор задачи «Шаблонная задача»

Зафиксируем число a_i. Заметим, что условие $a_i \cdot a_j \leqslant A$ эквивалентно условию $a_j \leqslant A/a_i$. Отсюда следует, что для каждого a_i нас интересует максимальное a_j, не превышающее A/a_j. Это уже стандартная задача. Её можно решать, например, так. Отсортируем a_i по возрастанию и будем идти по ним именно в таком порядке. Теперь оптимальное a_j можно искать двумя указателями или бинарным поиском.

Разбор задачи «Дораскладывать косынку»

Можно действовать жадно: каждый раз перекладывать минимальную по номиналу карту с ряда на подходящий дом. Очевидно, что таким способом мы корректно переложим все карты.