

## Задача А. Работа из дома

Отсортируем все отрезки из занятий, чтобы определить какие занятия придется проводить на полянке. Так как отрезки не пересекаются, а лишь касаются, сделать это можно по любой границе. Сложим в стек индексы очных занятий, чтобы определить какие онлайн занятия придется провести на полянке. Вытаскивая элемент из стека, рассмотрим его соседей в массиве, если они еще не лежали в стеке, но при этом невозможно успеть дойти до дома за перерыв, пометим их очными. Теперь, зная какие занятия очные, а какие нет, пройдем одним указателем. Для подсчета ответа удобно рассматривать концы занятия.

Если кончается онлайн занятие, а следующее очное, то к ответу добавим разность между началом второго занятия и концом первого минус  $t$  — время перехода. Если кончается онлайн занятие, и следующее онлайн, то к ответу добавится разность между концом второго занятия и концом первого. (он все время сидит дома) Если кончается очное занятие, а следующее онлайн, то к ответу добавится разность между началом второго занятия и концом первого минус  $t$  — время перехода. Если кончается очное занятие, и следующее очное, то Сид может попробовать сбегать до дома и назад, то есть, к ответу добавится разность между началом второго занятия и концом первого минус  $2 \cdot t$  — время перехода (если получившееся время положительно). Если это время отрицательное, то не будем его прибавлять.

Стоит не забыть также, что изначально он находится дома, а также после всех пар, возможно, успеет там отдохнуть. Поэтому можно в массив пар добавить нулевую пару и пару конца дня.

## Задача В. Орехнительная строка

*Автор задачи: Николай Будин, разработчик: Даниил Орешников*

Частичное решение, до которого можно было быстро догадаться — динамическое программирование. Можно было посчитать  $dp_{k,i,j}$  — минимальное количество ходов, которое необходимо сделать, чтобы собрать первые  $k$  орехов для коллекции и оказаться в клетке с координатами  $(i, j)$ . Для пересчета динамики можно перебрать в каком месте  $(i_0, j_0)$  был собран  $k$ -й орех, и обновить значение через  $dp_{k-1,i_0,j_0} + |i - i_0| + |j - j_0|$ . Такое решение работает в худшем случае за  $\mathcal{O}(n^2 m^2 |s|)$  (например, если почти все клетки материка заняты двумя видами орехов), и поэтому может пройти только первые две группы тестов.

Решить задачу на полный балл можно было с использованием поиска в ширину. Сначала построим граф переходов между клетками материка: рассмотрим  $nm$  вершин, и сопоставим каждой вершине свою клетку материка, после чего проведем ребра только между вершинами, соответствующими соседним клеткам. Затем сделаем  $|s|$  одинаковых копий полученного графа, которые мы назовем «слоями» и пронумеруем от 1 до  $|s|$ . Добавим ребра между слоями следующим образом: проведем ребро из вершины на слое  $k$ , соответствующей клетке с координатами  $(i, j)$ , в вершину на слое  $k + 1$ , соответствующую той же клетке материка, если  $x_{i,j} = s_k$ .

Теперь мы получили граф, в котором можно перемещаться внутри слоя, как на материке, и вперед на один слой, если в текущей клетке материка можно собрать следующий орех для коллекции. Сделаем на полученном графе поиск в ширину, тем самым найдем кратчайшие расстояния от стартовой вершины до всех остальных. Осталось только найти минимум расстояний по всем вершинам в последнем слое, в которых можно собрать последний орех, то есть в вершинах на слое  $|s|$  с координатами  $(i, j)$  такими, что  $x_{i,j} = s_{|s|}$ . В ответ следует вывести найденное расстояние, уменьшенное на  $|s| - 1$ , так как сбор орехов не занимает времени, а значит перемещения между слоями в ответе учитывать не надо.

Важно также заметить, что последняя группа тестов могла не проходиться решениями, которые строили граф в явном виде, потому что это требует большого количества памяти. Вместо этого стоило перебирать соседей текущей рассматриваемой вершины, пользуясь тем, что они за  $\mathcal{O}(1)$  определяются по номеру слоя и координатам текущей вершины внутри слоя. Время работы такого решения —  $\mathcal{O}(nm|s|)$ .

## Задача С. Игра с массивом

Заметим, что по каждому биту можно решать задачу независимо, а так как  $10^8 < 2^{27}$ , то ненулевыми битами могут быть только младшие 27.

Если рассмотреть один бит, задача сводится к следующей: Есть массив из 0 и 1, в нём изменяются какие-то элементы, и на подотрезке спрашивают, какое количество подотрезков с нечётным количеством единиц.

Эту информацию можно поддерживать с помощью дерева отрезков. Для того, чтобы "склеить" два отрезка, надо хранить на нём количество подотрезков с нечётным числом единиц, количество префиксов и суффиксов с нечётным числом единиц, длину отрезка и общее количество единиц. Итоговая асимптотика  $O(m \cdot \log n \cdot \log \max A)$ .