

Симулятор студента

Автор задачи и разработчик: Александр Гордеев

Сразу заметим, что есть ситуации, в которых код прожимать невыгодно. Например, если при нажатии кода наше суммарное перемещение гарантированно оставит нас дальше от конечной точки, чем та, в которой мы начинали. Остается только аккуратно проанализировать случаи, которые двигают нас в нужных направлениях. Из таких есть случаи, в которых мы по обеим координатам двигаемся в нужную сторону, и случаи, в которой по одной координате приближаемся, а по другой отдаляемся.

Обозначим за d_x и d_y суммарное перемещение по осям X и Y , соответственно, при нажатии кнопок, входящих в код. Чтобы их посчитать, достаточно проитерироваться по всем символам кода и учесть вклад каждого из них.

Ограничения первой подгруппы позволяли просто перебрать каждое действие. Во второй подгруппе можно было написать поиск в ширину (**bfs**). Из каждой клетки можно либо за стоимость 1 перейти в любую соседнюю, либо за стоимость $|S| + 1$ перейти в любую на расстоянии не больше d от текущей плюс $\overrightarrow{(d_x, d_y)}$. Таким образом, можно было в явном виде построить граф и запустить $(1 - s) - \text{bfs}$, где $s = |S| + 1$.

Теперь заметим, что если мы k раз нажмем код, то мы переместимся в клетку $(k \cdot d_x, k \cdot d_y)$. Если после этого не использовать код, то у нас останется $k \cdot d$ «бесплатных» перемещений благодаря коду, а остальные шаги придется сделать самостоятельно. Чтобы из такой клетки добраться до (x, y) , нужно $|x - k \cdot d_x| + |y - k \cdot d_y|$ шагов, поэтому ответом будет $\max(0, |x - k \cdot d_x| + |y - k \cdot d_y| - k \cdot d)$.

Количество нажатий кнопок в оптимальном ответе не может превышать $|x| + |y|$, поэтому в следующей подзадаче можно было перебрать k — количество нажатий кода, после чего для каждого фиксированного k получить указанную выше величину, и по всем таким выбрать минимум.

В группе с $|S| = 1$ можно было заметить, что нажатие кода перемещает нас только на 1 относительно текущей клетки. Поэтому при $d \leq 1$ вместо нажатия кода всегда можно просто нажать две кнопки движения, при $d \geq 3$ нажатие кода всегда не менее выгодно, чем два движения, а при $d = 2$ все зависит от того, в какую сторону нас двигает первая кнопка кода. Если в сторону от цели, то код всегда невыгоден, а иначе код может быть выгоден определенное число раз. Например, если $0 < y \leq 2x$, и код двигает нас на 1 вправо, выгодно $\lceil \frac{x+y}{3} \rceil$ раз нажать код, и мы придем в цель. Если же $y > 2x$, то выгодно только x раз нажать код, а дальше двигаться вверх простыми перемещениями. Остальные ситуации симметричны.

Для полного решения было необходимо внимательно рассмотреть полученную формулу. Заметим, что при увеличении k значение функции $(n + 1) * k + \max(0, |x - k \cdot d_x| + |y - k \cdot d_y| - k \cdot d)$ сначала уменьшается, потом увеличивается (ведь код перестаёт быть полезным когда мы уже достаточно вблизи клетки). А значит мы можем просто запустить тернарный поиск по положительным значениям и найти точку в которой значение минимально.

Для того чтобы решить задачу за $O(1)$ достаточно заметить, что достаточно найти минимальное k , такое что $f(k) = |x - k \cdot d_x| + |y - k \cdot d_y| - k \cdot d$ меньше нуля. Для этого достаточно перебрать все варианты раскрытия модули и из каждого варианта вычислить k , если $f(k) = 0$. Получили, что все возможные k имеют вид $\frac{a \cdot x + b \cdot y}{c \cdot d_x + d \cdot d_y + e \cdot d}$, где $a, b, c, d, e \in [-1 : 1]$. Перебираем все возможные дроби, каждую такую дробь, при её существовании, округляем вниз и вверх (ведь нас интересуют лишь целые числа) и подставляем значение в формулу, после чего берём минимум по всем $f(k)$.