Разбор задачи «Перекраска N-угольника»

Автор задачи — В. Гуровиц, автор разбора — М. Тихомиров

1 Формализация условия

Заметим несколько фактов относительно условия:

1. Внутри каждого четырехугольника триангуляции проведена ровно одна диагональ.

2. Условие об инвертировании диагонали внутри четырехугольника не влияет на алгоритм решения задачи.

3. Порядок перекрашивания не имеет значения, ведь среди перекрашиваемых диагоналей не может быть двух соседних (принадлежащих общему треугольнику триангуляции).

Заметим, что по 1 для каждой диангонали AC можно однозначно восстановить четырехугольник $ABCD$, в котором она проведена. Всвязи с этим применение операции, описанной в условии, к $ABCD$ будем называть просто *флипом* AC.

2 Идея решения

Отныне все стороны и диагонали триангуляции будем называть просто «диагоналями». Рассмотрим одну из них, p. Она разбивает данный многоугольник на левую (X) и правую (Y) части . Если в текущий момент мы флипнем p, то дальшейшая покраска будет проходить в X и Y независимо друг от друга. На этом основана идея решения — динамическое программирование на подмногоугольниках триангуляции, получаемых отрезанием от исходного вдоль какой-либо одной диагонали. Перед формализацией сказанного надо обсудить эффективный способ нумерации объектов исследования.

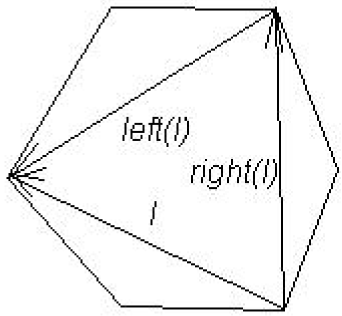
Рассмотрим диагональ l, соединяющую вершины с номерами i и j ($i \neq j$). Если идти от i к j, то по левую руку будет подмногоуольник, который мы обозначим S_1, а по правую — S_2 (один из них может быть тривиальным). Наоборот, если идти от j к i, то по левую будет S_2, по правую — S_1. Следовательно, каждый из нужных подмногоугольников однозначно соответствует ориентированной диагонали.

Теперь сформулируем две задачи.

1. Пусть дана некоторая триангуляция N-угольника и выделена одна из ориентированных диагоналей (напомним, что стороны также считаются диагоналями). Все стороны и диагонали изначально черные. Необходимо определить минимальное количество перекрашиваний, необходимых для того, чтобы полностью покрасить весь участок N-угольника справа от выделенной диагонали в красный цвет (S_2), либо определить, что это невозможно. Отметим, что флипать можно только внутренние диагонали S_2.

2. То же самое, но выделенная диагональ изначально красная.

Ответы на эти задачи обозначим соответственно $a(l)$ и $b(l)$, где l — выделенная диагональ. Также введем обозначения $left(l)$ и $right(l)$, которые соответствуют ближайшим справа от l ориентированным диагоналям, имеющим соответственно общий конец и начало.



Рассмотрим задачу $a(l)$. Поскольку сторону l по условию флипать нельзя, то чтобы она не осталась черной, надо флипнуть ровно одну из диагоналей $left(l)$ или $right(l)$ (одновременно обе нельзя, так как они соседние). Рассмотрим первый случай, тогда этим ходом станут красными $left(left(l))$, $right(left(l))$, $right(l)$ и l. Процессы дальшейшего перекрашивания подногоугольников, соответствующих первым трем диагоналям, никак не влияют друг на друга, поскольку они не пересекаются по внутренним диагоналям. Поэтому надо взять минимум шагов по каждой из этих подзадач, получим $b(left(left(l)))+b(right(left(l)))+b(right(l))$. Аналогичным образом рассматривается второй случай.

Получили рекуррентную формулу для $a(l)$:

$$a(l) = min(b(left(left(l))) + b(right(left(l))) + b(right(l)),\\ b(left(right(l))) + b(right(right(l))) + b(left(l)))) + 1$$

Теперь рассмотрим задачу $b(l)$. Если в минимальном решении флипается одна из $left(l)$ или $right(l)$, то ответ совпадает с $a(l)$. Если ни одна из этих диагоналей не используется, то ответ равен $a(left(l)) + a(right(l))$. Окончательная формула:

$$b(l) = min(a(l), a(left(l)) + a(right(l)))$$

3 Реализация

Напомним, что стороны по-прежнему считаются диагоналями.

Решим задачу с использованием метода динамического программирования.

Сначала предподсчитаем $left(l)$ и $right(l)$ для всех l. Для этого используем сортировку всех ориентированных диагоналей. Для подсчета $left(l)$ сортировка идет по концам диагоналей, а при равных — по началам в порядке обхода многоугольника против часовой стрелки. Пусть в полученном списке элемент b идет сразу после a. Если если их концы совпадают, то $left(b) = a$. В противном случае тогда b — сторона многоугольника, ориентированная согласно обходу против часовой стрелки, и $left(b)$ не определено. Аналогично вычисляется $right(l)$.

Далее организуем порядок вычислений. Поскольку задача сводится к меньшим размерностям, отсортируем ориентированные диагонали по количеству вершин в соответствующих им участкам многоугольника — в правых подмногоугольниках. После этого вычисления происходят в порядке увеличения числа вершин: сначала вычисляется $a(l)$, затем $b(l)$. Инициализация: если l — сторона против часовой стрелки обхода многоугольника, справа нет ничего и $a(l) = \infty$, $b(l) = 0$. При вычислениях надо помнить, что функциям a и b от неопределенных диагоналей соответствует бесконечность. Ответом на задачу будет являться $a(l)$, где l — любая сторона многоугольника по часовой стрелке обхода многоугольника.

Нетрудно видеть, что сложность работы алгоритма равна $O(N \log N)$.