Задание №85. $MaximumRectangle$

Постановка задачи

Дана бинарная матрица $\mathrm{matr}[n \times m]$, состоящая из символов '0' и '1'. Требуется найти площадь **максимального прямоугольника**, полностью состоящего из единиц.

Обзор алгоритма

Алгоритм использует динамическое программирование с предобработкой:

1. Предобработка: для каждой ячейки (i, j) вычисляем $\mathrm{info}[i][j]$ – количество подряд идущих единиц слева в строке i, заканчивающихся в j.
2. Динамическое программирование: $\mathrm{dp}[i][j]$ хранит максимальную площадь прямоугольника из единиц в подматрице $(0, 0) - (i, j)$.
3. Для каждой ячейки (i, j), рассматриваемой как правый нижний угол, перебираем все возможные верхние грани k и вычисляем площадь прямоугольника.

Детали реализации

- $\mathrm{info}[i][j] = \begin{cases} 0, & \text{если } \mathrm{matr}[i][j] =' 0' \\ \mathrm{info}[i][j-1] + 1, & \text{если } \mathrm{matr}[i][j] =' 1' \end{cases}$
- $\mathrm{dp}[i][j] = \max(\mathrm{dp}[i-1][j], \mathrm{dp}[i][j-1], S_{\max}(i, j))$, где $S_{\max}(i, j)$ – максимальная площадь прямоугольника, заканчивающегося в (i, j).
- Для вычисления $S_{\max}(i, j)$ перебираем все k от i до 0:

$$\mathrm{height} = i - k + 1, \quad \mathrm{width} = \min_{t=k}^{i} \mathrm{info}[t][j], \quad S = \mathrm{height} \times \mathrm{width}$$

Доказательство корректности

Определение 1. *Прямоугольник R в матрице задаётся координатами (x_1, y_1) (левый верхний) и (x_2, y_2) (правый нижний). Площадь $S(R) = (x_2 - x_1 + 1) \times (y_2 - y_1 + 1)$.*

Лемма 1 (О представимости прямоугольника)**.** *Любой прямоугольник R, состоящий только из единиц, однозначно определяется:*

1. *Его правым нижним углом (i, j)*
2. *Верхней границей k (номер строки, где начинается прямоугольник)*
3. *Шириной $w = \min_{t=k}^{i} info[t][j]$*

Доказательство. Для прямоугольника из единиц в каждой строке $t \in [k, i]$ должен быть непрерывный отрезок единиц длины хотя бы w, заканчивающийся в столбце j. Максимальная возможная ширина определяется минимальным значением $\text{info}[t][j]$ по всем строкам от k до i, так как прямоугольник должен быть шириной не более этого значения во всех строках. □

Теорема 1 (Полнота перебора)**.** *Алгоритм перебирает все возможные прямоугольники из единиц.*

Доказательство. Рассмотрим произвольный прямоугольник R из единиц с координатами $(k, j - w + 1)$ до (i, j), где w – ширина прямоугольника. Алгоритм при обработке ячейки (i, j) в цикле по k от i до 0 рассмотрит данный прямоугольник:

- При $k = x_1$ (верхняя граница R)
- На каждой итерации поддерживается $\text{ry} = \min\{\text{info}[t][j] \mid t \in [k, i]\}$
- Когда $\text{ry} \geq w$, площадь $(i - k + 1) \times \text{ry}$ будет не меньше площади R
- Максимум по всем k даст площадь не меньше $S(R)$

Таким образом, каждый прямоугольник учтётся при вычислении $S_{\max}(i, j)$ для его правого нижнего угла. □

Теорема 2 (Корректность динамического программирования)**.** *$dp[i][j]$ содержит максимальную площадь прямоугольника из единиц в подматрице* $(0, 0) - (i, j)$.

Доказательство. Доказательство по индукции.

База: Для $\text{dp}[0][0]$:

$$\text{dp}[0][0] = \begin{cases} 1, & \text{если } \text{matr}[0][0] =' 1' \\ 0, & \text{иначе} \end{cases}$$

Это корректно, так как максимальный прямоугольник в подматрице 1×1 либо имеет площадь 1, либо 0.

Шаг индукции: Предположим, $\text{dp}[i-1][j]$ и $\text{dp}[i][j-1]$ корректны. Максимальный прямоугольник в $(0, 0) - (i, j)$ может:

1. Не содержать (i, j): тогда он полностью лежит в $(0, 0) - (i - 1, j)$ или $(0, 0) - (i, j - 1)$
2. Содержать (i, j) как правый нижний угол

Случай 1 учитывается максимумом из $\text{dp}[i-1][j]$ и $\text{dp}[i][j-1]$. Случай 2 учитывается вычислением $S_{\max}(i, j)$, которое по Лемме 1 и Теореме 1 находит максимальную площадь прямоугольника, заканчивающегося в (i, j).

Таким образом, $\text{dp}[i][j] = \max(\text{dp}[i-1][j], \text{dp}[i][j-1], S_{\max}(i, j))$ корректно.

После заполнения всей матрицы dp, значение $\text{dp}[n-1][m-1]$ содержит площадь максимального прямоугольника из единиц во всей матрице. □

Доказательство. Следует из Теоремы 2, так как $(n-1, m-1)$ является правым нижним углом всей матрицы. $\square$

Анализ сложности

- **Предобработка:** $O(n \cdot m)$
- **Вычисление $S_{\max}(i, j)$:** В худшем случае $O(n)$ для каждой ячейки
- **Общая сложность:** $O(n^2 \cdot m)$
- **Память:** $O(n \cdot m)$ для хранения dp и info

Оптимизации и замечания

1. Внутренний цикл по k можно прервать, когда $\mathrm{ry} = 0$
2. Алгоритм можно оптимизировать до $O(n \cdot m)$ с использованием стека (алгоритм максимальной площади в гистограмме, применённый построчно)
3. Текущая реализация проще для понимания и демонстрирует основные идеи динамического программирования

Пример корректности

Рассмотрим матрицу:

$$\begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Шаг 1: info:

$$\begin{bmatrix} 1 & 0 & 1 & 2 \\ 1 & 2 & 3 & 0 \\ 1 & 2 & 3 & 4 \end{bmatrix}$$

Шаг 2: При вычислении dp[2][3]:

- Рассматриваем $k = 2$: $\mathrm{ry} = 4$, площадь $1 \times 4 = 4$
- $k = 1$: $\mathrm{ry} = \min(4, 0) = 0$, прерываем
- $k = 0$ не рассматриваем из-за прерывания
- Максимум из dp[1][3], dp[2][2] и найденных площадей даст 6

Максимальный прямоугольник площадью 6 найден корректно.