КМП-алгоритм. Z-алгоритм

Минский ШАД. Весна

20 апреля 2015 г.

1 Обозначения

Бордером строки будем называть такой её префикс, который совпадает с суффиксом. Например, у строки «abacaba» бордерами являются «a», «aba» и «abacaba». **Собственным бордером** будем называть бордер, меньший по длине, чем сама строка.

π-последовательностью строки S будем называть числовой вектор π размера $|S|$, такой что π_i равен длине самого длинного собственного бордера первых $i+1$ символов строки (в ноль-индексации). Например, для «abacaba» $\pi = \{0, 0, 1, 0, 1, 2, 3\}$.

z-последовательностью строки S будем называть числовой вектор z размера $|S|$, такой что z_i равен длине максимальной (по длине) подстроки, которая начинается в позиции i и совпадает с некоторым собственным суффиксом строки. Например, для «abacaba» $z = \{0, 0, 1, 0, 3, 0, 1\}$.

2 Тематические задачи

1. [½ балла] Дана числовая последовательность. Проверить, правда ли, что существует строка, π-последовательность которой совпадает с данной последовательностью.

2. [½ балла] Дана π-последовательность строки. Предложить алгоритм поиска любой строки, порождающей данную последовательность, и доказать корректность полученного алгоритма (вообще всегда надо доказывать, но тут я подчеркну).

3. [½ балла] Дана изначально пустая строка. Каждый ход к ней дописывается один символ в начало или в конец. После каждого хода за $\mathcal{O}(|S|)$ необходимо говорить, сколько различных подстрок существует в S. Предполагается решение с помощью КМП или z-алгоритма.

4. [½ балла] Предложить алгоритм поиска второго по длине собственного бордера данной строки. Сложность должна быть $\mathcal{O}(|S|)$.

   Решение:

   Вычислим префикс функцию для строки с помощью алгоритма Кнута-Морриса-Прата и ответом тогда будет являться $\pi_{\pi_{|S|}}$

5. [½ балла] Предложить алгоритм вычисления количества различных бордеров у строки S за время $\mathcal{O}(|S|)$.

   Решение:

   По определению $\pi_{|S|}$ — длина максимального бордера строки S. $\pi_{\pi_{|S|}}$ — второго и так далее. Построим дерево, где каждая вершина будет соответствовать позиции в строке (или, что эквивалентно, длине бордера). Тогда проведём ребро из вершины i в вершину π_i. Очевидно, что ответ на задачу, это глубина вершины с номером $|S|$. Само дерево, конечно можно не строить, достаточно

лишь поддерживать в массиве высоту очередной вершины, т.е. при вычислении π_i выполнять $h_i \leftarrow h_{\pi_i} + 1$.

6. [½ балла] Для каждой позиции строки S вычислить значение a_i — длину максимальной подстроки, которая начинается в i и совпадает с некоторым суффиксом строки S. Решение должно иметь сложность $\mathcal{O}(n)$.

Решение:

Развернём строку и посчитаем префикс-функцию. Если мы развернём обратно массив, содержащий значения префикс-функций, то можно заметить, что это и есть ответ на задачу.

7. [1 балл] Дана строка S, пусть $|S| = n$. Затем следуют запросы вида (i, j), на каждый из которых надо ответить длину j-го бордера (считая бордеры упорядоченными по длине) у строки, которая является i-префиксом строки S, т.е. у подстроки $S[0 \dots i-1]$. На каждый запрос следует отвечать не медленней, чем за $\mathcal{O}(\log n)$. Разрешается сделать препроцесс за $\mathcal{O}(n \log n)$.

Решение:

Построим дерево из решения предыдущей задачи. Тогда если мы хотим ответить на вопрос, то нам надо сделать из вершины i ровно $j-1$ шаг вверх. Так как j может быть большим, сделаем предпросчёт такого вида: $f_{i,k}$ — в какую вершину дерева мы попадём, если сделаем из вершины i ровно 2^k шагов вверх. Очевидно, что $f_{i,0}$ — просто отец вершины i. С другой стороны $f_{i,k} = f_{f_{i,k-1},k-1}$. Теперь, чтобы посчитать ответ, достаточно разложить число j на сумму степеней двойки и сделать соответсвующие шаги.

8. [1 балл] Рассмотрим следующую игру для двух игроков. Первый игрок загадывает строку S и сообщает второму её длину n. Также у первого игрока есть изначально пустая строка T. Затем игроки ходят по очереди, начиная с первого. На своём ходу первый игрок добавляет в конец строки T любую букву (строка T также неизвестна второму игроку). Второй игрок имеет право задать первому не более пяти вопросов вида «правда ли, что такой-то символ строки S (T) совпадает с таким-то символом строки S (T)». Т.е. можно сравнивать любые позиции в одной и той же строке либо в разных строках. Единственное ограничение — нельзя задавать более пяти вопросов за ход. В конце своего хода второй игрок обязан сказать, сколько на данный момент подстрок строки T совпадают со строкой S. Ваша задача разработать такую стратегию игры для второго игрока, чтоб его ответ всегда был правильным вне зависимости от игры первого игрока, либо доказать, что такой стратегии не существует.

Решение:

Можно переформулировать вопрос, на который должен ответить второй игрок следующим образом: правда ли, что в новом символе строки T заканчивается какое-либо вхождение строки S. Действительно, тогда на каждом ходу стоит лишь отвечать сколько вхождений уже было.

Мы вроде бы знаем, что КМП-алгоритм не является хорошим интерактивным алгоритмом, хоть в общем он делает не более $2n$ сравнений для строки длины n, но на вычисление некоторых значений π_i может потребоваться линейное количество времени. Однако мы увидим, что это нестрашно.

Заметим, что максимальное количество сравнений, которое может сделать алгоритм КМП на строке длины n равно $2n-1$. А это значит, что мы успеем посчитать π-последовательность для строки S, пока в строки S только $\approx \frac{2}{5}|S|$ символов, а значит ещё ни одного вхождения быть не могло. Более того, в худшем случае, первое вхождение может случиться, когда $|S| = |T|$, но за это время нам уже дадут посчитать $5|S|$ сравнений, хотя нам нужно только $4|S|$.

Теперь заметим следующий факт. Пусть мы вычисляем префикс-функцию в позиции i и нам не хватило 5 сравнений на это (например такое возможно в строке «aaaaaaaaaaaaab» на символе «b»). Но тогда мы знаем, что π_i как минимум на 5 меньше, чем $|S|$, а значит вхождение строки S не может начинаться ни в этом символе, ни в последующих пяти символах.

Таким образом, чтобы решить задачу нужно просто выполнять КМП, прерываясь каждое пятое сравнение. Потратив чуть времени, можно показать, что мы всегда будем успевать находить вхождение за нужное время (конечно, от вас я ждал такого доказательства).

9. [1 ½ балла] Дана π-последовательность строки. Посчитать количество строк, которые порождают данную последовательность над алфавитом размера C. Сложность алгоритма должна составлять $\mathcal{O}(|\pi| \log |\pi|)$. Можно считать, что искомое количество помещается в машинное слово (тем не менее, это не значит, что от него должна зависеть сложность).

10. [1 ½ балла] Задана строка, пожатая RLE-алгоритмом, т.е. последовательностью пар (c_i, l_i) — символ и количество повторений соответственно. Например, строка «aaabbaeee» будет закодирована такой последовательностью: $\{(a,3),(b,2),(a,1),(e,3)\}$. Таких пар — N. Также заранее задано M вопросов вида «каково значение префикс-функции данной строки в позиции i?». Предложить алгоритм ответа на эти запросы за время $\mathcal{O}(T \log^2 T)$, где $T = \max N, M$.

11. [1 балл] Доказать или опровергнуть следующие утверждения (π – π-последовательность, z — z-последовательность):
    (a) $\sum \pi_i = \sum z_i$ для любой строки
    (b) $\sum \pi_i < \sum z_i$ для любой строки
    (c) $\sum \pi_i > \sum z_i$ для любой строки
    (d) $\sum \pi_i \leqslant \sum z_i$ для любой строки
    (e) $\sum \pi_i \geqslant \sum z_i$ для любой строки
    (f) Существует строка, что $\pi_i > z_i$ для любого $i > 0$
    (g) Существует строка, что $z_i > \pi_i$ для любого $i > 0$

3 Задачи на повторение

12. [½ балла] На прямой своими координатами задано n точек. В этих точках расположены гвоздики. Два гвоздика, находящихся в позициях x_i и x_j, можно соединить ниткой длиной $|x_i - x_j|$ саженей. Необходимо натянуть нитки между гвоздями таким образом, чтоб к каждому гвоздю была присоединена как минимум одна нитка, а суммарная длина нитей была минимальна. Сложность алгоритма должна составлять $\mathcal{O}(n \log n)$.

Решение:

Отсортируем все гвоздики по координате и будем считать, что они пронумерованы в порядке увеличения координаты. Очевидно, что гвоздик стоит соединять только с соседним гвоздём (иначе можно считать что рассматриваемый гвоздь соединён с промежуточным, а промежуточный — с изначальным соседом). Тогда введём величину f_i — ответ на задачу, если бы было задано только первых i гвоздей. Тогда:

$$\boxed{f_i = \min(f_{i-1}, f_{i-2}) + |x_i - x_{i-1}|}$$

Последний гвоздь мы обязаны соединить с предпоследним. Мы выбираем из двух вариантов: первый соответствует случаю, когда мы соединяем гвоздь $i-1$ с гвоздём $i-2$, а второй — нет. Итого $\mathcal{O}(n \log n)$ на сортировку и $\mathcal{O}(n)$ на вычисление ответа.

13. [½ балла] Дано N пар натуральных чисел (a_i, b_i). Необходимо посчитать количество различных пар натуральных чисел (A, B), таких, что $\exists i : A \leqslant a_i \wedge B \leqslant b_i$, за время $\mathcal{O}(N \log N)$.

4 Практические задачи

Ссылка на контест: `https://contest.yandex.ru/contest/1080/problems/`

14. [1 балл] Реализовать решение задач 1 и 2.

15. [1 балл] По π-последовательности строки найти её z-последовательность (используй 2, Люк).

Задание	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Сумма
Баллы	½	½	½	½	½	½	1	1	1½	1½	1	½	½	1	1	12