Алгоритм Ахо-Корасик. Суффиксные структуры

Минский ШАД. Весна

20 апреля 2015 г.

1 Примечания

Напоминаю, что суффиксный массив человечество умеет строить за линейное время от длины строки, чем мы и будем пользоваться. Тем не менее, ввиду сложности данного алгоритма, в практической части разрешается использовать алгоритм построения за $\mathcal{O}(n \log n)$.

2 Тематические задачи

1. [1 ½ балла] Маленькому Сэмюэлю на день рождения подарили набор A из N кодовых слов, причём суммарная длина всех слов равна L. Он хочет проверить, правда ли, что набор задаёт однозначно декодируемый код, т.е. из того, что $a_1 a_2 \dots a_n = b_1 b_2 \dots b_m$, где $\forall i : a_i \in A, b_i \in A$, следует, что $n = m$ и $\forall i : a_i = b_i$.

   У малыша не так много времени, поэтому алгоритм должен иметь сложность $\mathcal{O}(LN)$.

Решение:

Сэмюэль Морзе является автором одного известного неоднозначно декодируемого кода (для однозначности используется доп. символ — пауза).

Попробуем построить такие две последовательности кодовых слов так, чтобы из конкатенация совпала, хотя сами последовательности различались. Можно считать, что такие две последовательности начинаются с разных кодовых слов (иначе можно откидывать одинаковые пары, пока не встретим различие).

Переберём эту пару кодовых слов (т.е. те слова, которые будут равны a_1 и b_1). Очевидно, что одно из них длинней другого. Теперь будем следовать такому алгоритму. Пока одна из последовательностей короче (как конкатенация слов), чем другая, будем подбирать в более короткую такое слово, что его добавление не испортит равенства строк.

К примеру пусть на каком-то этапе алгоритма у нас получилась такая ситуация:

•	•	•	б	р	е	с	т
•	а	м	б	р	е	?	?

В такой ситуации в конец более короткой последовательности мы можем приписать слова «стратосфера», «стагнация», «стоп» или «с», но не можем «сила», «есть», «ума», «не» или «надо».

Если можно выбрать несколько вариантов, то будем пробовать все (т.е. воспользуемся перебором с возвратом).

Если мы в какой-то момент смогли сравнять длины двух строк, то мы можем сказать, что код не однозначно декодируемый, иначе однозначно.

Конечно, такой алгоритм не будет удовлетворять заданному временному ограничению. Более того, он может работать бесконечно долго. Заметим, однако, что всё, что нас интересует на каждом

шаге перебора, это какой суффикс остался от более длинной строки. Понятное дело, что если на какой-то момент мы получили суффикс, который уже хоть раз получали — то дальше углубляться в перебор не стоит.

Каждый суффикс более длинной строки — это суффикс одного из кодовых слов. Различных суффиксов не более L. Давайте заведём граф, где каждому из таких суффиксов поставим в соответствие веришну. Из каждой вершины попробуем провести дуги, которые будут соответствовать переходу в переборе. Т.е. мы должны перебрать слово, которое мы хотим приписать к более короткой строке и понять, правда ли, что его префикс совпадает с текущим суффиксом (либо он сам совпадает с префиксом этого суффикса). Понятно дело, что на такие запросы мы можем отвечать за $\mathcal{O}(1)$, построив предварительно суффиксный массив и LCP.

Единственная детали, что мы не можем позволить себе делать sparse-table в этой задаче, так как тогда сложность будет $\mathcal{O}(NL + L \log L)$. Заметим однако, что мы можем предпросчитать все ответы по построенному LCP заранее. Действительно просто найдём все наши кодовые слова и за линейный проход по массиву LCP получим ответы на LCP-запросы для каждого суффикса.

2. [1 балл] Маленький Джордж загадал бинарную строку из N бит. Он посчитал по ней суффиксный массив и отдал вам его. Сможете ли вы отгадать строку, которую загадал малыш? Кстати говоря, он мог ошибиться и предоставить вам массив, которому не соответствует ни одна бинарная строка, тогда надо указать ему на ошибку. Тем не менее, малыш умён не по годам, поэтому если вы назовёте строку, которая порождает такой же массив, как и загаданная, то он по доброте душевной сочтёт загадку разгаданной.

   Если вы будете решать загадку слишком долго, малыш решит, что вы очень скучный человек, и пойдёт к студентам киевского филиала, так что решите задачу за $\mathcal{O}(N)$.

Решение:

Джордж Буль — самый известный математик, работавший в сфере мат. логики. В честь него назван булевый тип данных.

Заметим, что если строка состоит из одних нулей или одних единиц, то массив будет выглядить $(n-1, n-2, \ldots, 0)$. Так что если на входе такой массив, то сразу выдадим ответ.

Иначе предположим, какой мог быть последний бит строки. Если он был нулевым, то, очевидно, суффикс только из этого бита должен стоять на первом месте суффиксного массива (нет строки меньше, чем «0»). Можем откинуть этот бит и решать задачу для строки на единицу короче. Если же это единица, то это самый маленький (лексикографически) суффикс, который начинается на единицу. Тогда можем найти его позицию в суффиксном массиве, все суффикс до него начинаются с нуля, все после него — с единицы (т.е. мы восстановили строку).

Осталось проверить эту строку. Можно просто построить суффиксный массив за линейное время и сравнить его с данным (этот шаг существеннен).

3. [½ балла] Малыши Майкл и Ричард играют в следующую игру. Майкл придумал строку из S символов и дал Ричарду примерно $\mathcal{O}(|S| \log |S|)$ времени, чтобы вдоволь её изучить. После этого он задаёт вопросы вида « правда ли, что в позиции i строки S начинается тандемный повтор длины k?». Другими словами, правда ли, что $S[i \ldots i+k-1] = S[i+k \ldots i+2k-1]$. Ричард не хочет опозориться перед Майклом (в будущем им, возможно, придётся вместе писать статьи), поэтому он просит Вас помочь отвечать на каждый такой вопрос за $\mathcal{O}(1)$.

Решение:

Майкл Майн и Ричард Лоренц впервые предложили алгоритм поиска всех тандемных повторов за $\mathcal{O}(n \log n)$

Построим массив LCP. Теперь нам просто надо отвечать, правда ли, что $LCP(i, i+k) \geqslant k$.

3 Задачи на повторение

4. [1 балл] Малышу Палу подарили массив из $nm+1$ различных чисел. Он знает, что не так интересен массив чисел, как возрастающие (ну или хотя бы убывающие) подпоследовательности. Однако малыш, имени которого, к сожалению, наш герой не разобрал, сказал ему, что тот никогда не найдёт в своём массиве возрастающей подпоследовательности длины $n+1$ и, уж тем более, убывающей подпоследовательности длины $m+1$. Малыш Иоганн тут же успокоил Пала, сказав, что это не так. Станьте номером один для Пала и докажите, что слова Иоганна не пустой звук, т.е. всегда найдётся хотя бы одна из указанных подпоследовательностей.

Решение:

В задаче требуется доказать теорему Пала Эрдёша и Дьёрдя Секереша. «Номер один» — конечно же, отсылка к числу Эрдёша.

Поставим числу под номером i пару (a_i, b_i) — длину самой длинной возрастающей и убывающей последовательностей, заканчивающихся в позиции i, соответственно. Понятно, что для любых $i < j:\ a_i < a_j \cup b_i < b_j$ (действительно число в позиции j либо больше, чем в позиции i, либо меньше, а значит всегда можно продлить хотя бы одну из последовательностей). Если предоложить, что злой малыш Дьёрдь прав, то всего разных пар бывает nm, но так как чисел $nm+1$, то по принципу Иоганна Петера Густава Лежёна Дирихле найдётся хотя бы одно, где $a_i > n$ либо $b_i > m$.

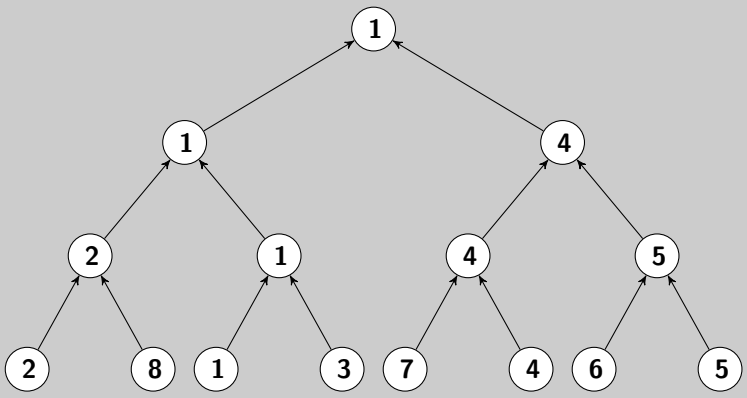
5. [1 балл] Малыш Чарльз подарил малышам Энтони и Ричарду массив из $n = 2^k$ различных целых чисел. А как известно, чем меньше числа, тем лучше. Энтони очень заботится о малыше Ричарде, поэтому уступает ему минимум из подаренного массива. Тем не менее, ему тоже очень хочется узнать, какой же подарок получит он. А значит, он хочет определить 2-ю порядковую статистику за не более, чем $n + k - 2$ сравнения (больше ждать он уж не в силах). Помогите ему составить алгоритм, который найдёт 2-ю порядковую статистику не более, чем за приведённое количество сравнений.

Решение:

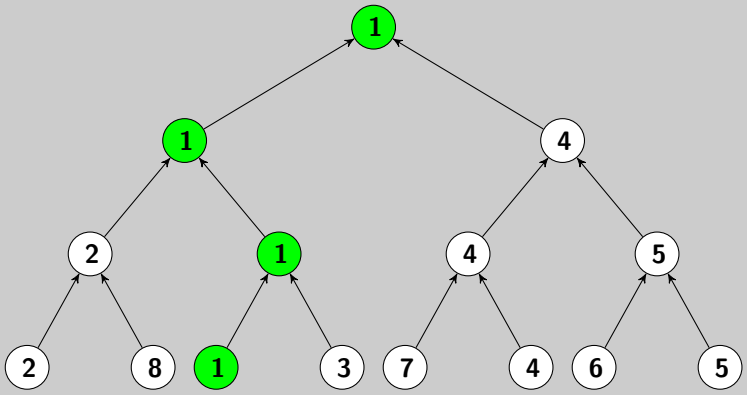
Чарльз Энтони Ричард — это всё имена Хоара.

Заметим, что найти минимум в массиве из n элементов можно только с помощью $n-1$ сравнения, причём, очевидно, меньше сделать нельзя.

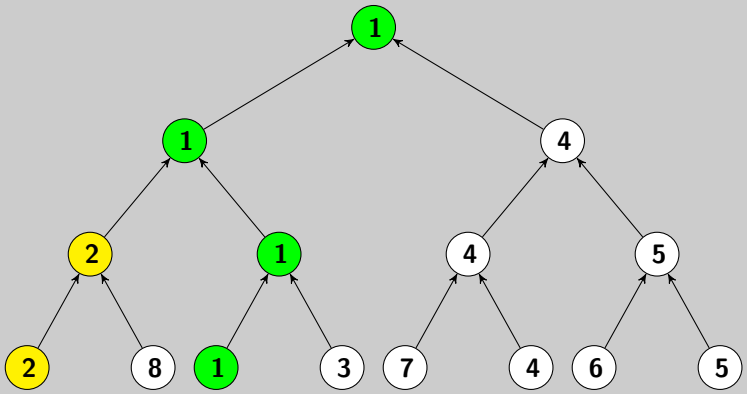
С другой стороны можно по разному использовать эти сравнения. Давайте будем поступать следующим образом. На первом шаге сравним элементы на первом и втором местах, затем на третьем и четвёртом и так за $n/2$ сравнений оставим ровно $n/2$ кандидатов на минимум. Будем повторять такую операцию, пока не останется ровно один элемент. Такую стратегию легко реализовать в виде дерева. К примеру, рассмотрим массив $(2, 8, 1, 3, 7, 4, 6, 5)$:



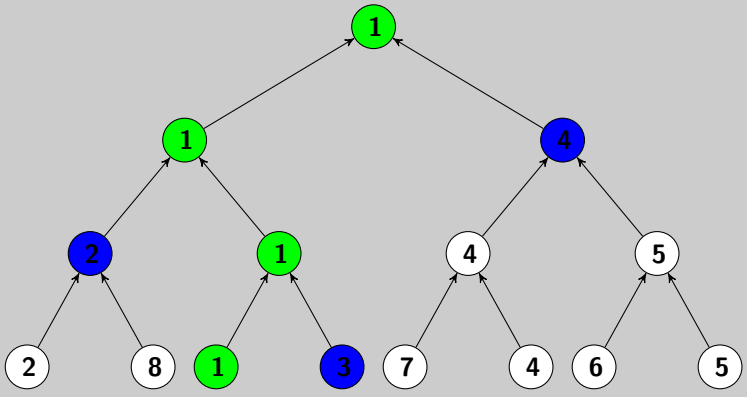
Посмотрим, как минимум проложил себе путь наверх:



Очевидно, он выигрывал при каждом сравнении. Теперь заметим, что среди тех, у кого он выигрывал обязательно есть второй минимум. Действительно, второй минимум таким же образом шёл наверх, выигрывая всех, пока не встретился с минимумом:



Заметим, что всего элементов, которых мы сравнивали с минимумом ровно k — по одному на каждый уровень:



А значит, из них мы можем найти минимум за $k-1$ сравнение. Таким образом нам нужно $n-1+k-1=n+k-2$ сравнения на всё.

4 Практические задачи

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

6. [1 балл] Реализуйте решение задачи 3.

7. [1 балл] Дана строка длины n и её суффиксный массив. Требуется найти количество различных строк в массиве, максимум в массиве LCP и количество различных значений функции $LCP(i, j)$ за $\mathcal{O}(n)$.

8. [1 балл] Малыш Абрахам получил от малыша Якоба в подарок строку. Его друг, малыш Терри, тут же решил загадать Абрахаму загадку. А именно, его интересует, как сильно похожа строка, начиная с позиции i, на какую-нибудь подстроку, начинающуюся в более ранней позиции. Более формально, хочется найти $f(i)$ для всех возможных i, где $f(i) = \max\{k \ : \ \exists j < i, s[j, j + k] = s[i, i + k]\}$.

9. [2 балла] Дано n строк общей длины T. Также дано m запросов вида «в скольки различных данных строках есть подстрока Q_i?». Строки Q_i образуют беспрефиксный код. Общий размер запросов не превышает S. Необходимо ответить на все запросы за время $\mathcal{O}(S + T)$.

Задание	1	2	3	4	5	6	7	8	9	Сумма
Баллы	1½	1	½	1	1	1	1	1	2	10