Домашнее задание №13
Алгоритмы. 5 курс. Весенний семестр.

Горбунов Егор Алексеевич

18 мая 2016 г.

Задание №1 Найти среднее lcp попарно всех суффиксов за $\mathcal{O}(n)$

Решение:

Задание №3 Найдите максимальный рефрен — такую подстроку строки s, что количество ее вхождений (возможно пересекающихся) помноженных на ее длину — максимально. Решите двумя способами: суффиксными деревом и массивом.

Решение:

(a) Строим по строке s суффиксное дерево T. Ответом на вопрос задачи будет какая-то позиция в этом дереве, т.к. любая позиция в T отвечает некоторой подстроке. Заметим, что ответ всегда выгодно искать именно в вершинах, а не где-то по среди ребра, т.к. вдоль ребра число вхождений подстроки не меняется, а значит, можно пройти вниз по ребру до ближайшего ветвления, что только увеличит длину подстроки (нас интересует макс. длина). Тогда предподсчитаем в вершинах T длины подстрок $height(v)$, которые в них заканчиваются (глубины вершин в смысле числа символов), а так же размер поддерева в каждой вершине $size(v)$, т.е. число вершин, которые находятся ниже её самой и для которых она предок. Такие штуки считаются обходом дерева, а значит за линейное время от числа вершин дерева, а значит за $\mathcal{O}(|s|)$. Теперь осталось ещё раз обойти дерево, и среди вершин выбрать ту, у кторой $size(v) * height(v)$ максимально. ∎

(b) Построим суффиксный массив A по строке s, а так же массив LCP. Если какая-то подстрока встречается в s k раз, то есть k суффиксов, которые идут подряд, будучи лексикографически отсортированы, а суффиксный массив как раз и задаёт такой порядок. Пойдём от обратного. Рассмотрим отрезок $[l, r)$: $\min_{i \in [l,r)} LCP[i]$ — это длина подстроки, которая встречается $r - l$ раз. Таким образом нам нужно найти:

$$(r - l) \cdot \min_{i \in [l,r)} LCP[i] \longrightarrow \max_{[l,r)}$$

Заведём массив B. Будем перебирать i в порядке убывания $LCP[i]$ и ставить 1 в ячейку $B[i]$. Теперь найдём в массиве B такую минимальную позицию $l \le i$, что $\sum(B[l..i]) = i - l + 1$ (т.е. такое l, что всё от l до i в массиве B заполнено 1, т.е. такие LCP-шки были уже добавлены). И аналогично найдём такую границу $r \geq i$. И обновим ответ $ans = \max(ans, (r - l + 1) \cdot \mathrm{LCP}[i])$. Сумму на отрезках B можно считать деревом отрезков, а границы l и r подбирать бинарным поиском. Итого получится алгоритм за $\mathcal{O}\left(n \log^2(n)\right)$. Это быстрее чем n^2, а поэтому можем считать использование суфф. массива оправданным :) ■

Задание №5 Даны k строк суммарной длины n. Найдите p-ю лексикографически общую их подстроку за $\mathcal{O}(n)$.

Решение: Будем решать суффиксным деревом. Рассмотрим строку $s_1\$s_2\$\ldots\$s_k\$\$$. (два доллара на конце — это важно). Её длина всё ещё $\mathcal{O}(n)$. Построим по такой строке суффиксное дерево T. Отбросим у этого суффиксного дерева ветку, которая идёт из корня по символу $. Теперь нам нужно найти такую вершину дерева T, что в поддереве этой вершины есть k листьев и при этом строка от корня до этой вершины p-ая в лексикографическом порядке. Для этого 1) нужно предподсчитать размеры поддеревьев и 2) обходить детей вершин в лексикографическом порядке при обходе в глубину. Всё. По-идее решение должно быть за $\mathcal{O}(n)$.