

## Домашняя работа.

### Строки.

#### Задание №1.

Условие:

1. Дана строка. После предподсчета за  $O(n)$ , отвечать за  $O(1)$  на запрос: чему равна хеш-функция подстроки  $s[l..r]$ ?

Решение:

Для начала за  $O(n)$  посчитаем все хеши для всех префиксов строки  $S$ . После этого чтобы за  $O(1)$  узнать равны ли хеши подстроки в строке и введенной строки будем из хеша последнего элемента в подстроке вычитать хеш первого элемента в подстроке и делить на число равное степени двойки по индексу первого элемента подстроки.

Пример:

$$h[i \dots j] = s[i] + s[i + 1] * 2^1 + s[i + 2] * 2^2 + \dots + s[j] * 2^j$$

$$h[i \dots j] = \frac{h[0 \dots j] - h[0 \dots i - 1]}{2^i}$$

$$I = \{0, 1, 2, 3, 4, 5, 6\};$$

$$s = \text{"abacaba"};$$

$$s_{min} = \text{"aba"};$$

$$s[4 \dots 6] ? s_{min}$$

$$h[0 \dots 6] = 1 * 1 + 2 * 2 + 1 * 4 + 3 * 8 + 1 * 16 + 2 * 32 + 1 * 64 = 1 + 4 + 4 + 24 + 16 + 64 + 64 = 177;$$

$$h[0 \dots 3] = 1 * 1 + 2 * 2 + 1 * 4 + 3 * 8 = 33;$$

$$h[4 \dots 6] = (177 - 33) / 16 = 9;$$

$$h[s_{min}] = 9;$$

хеши равны также как и строки.

#### Задание №2.

Условие:

2. Дана строка. Надо отвечать на запросы  $lcp(i, j)$ : длина наибольшего общего префикса строк  $s[i..n]$  и  $s[j..n]$ .  $O(\log n)$  на запрос.

Решение:

Берем идею из первого задания. Сначала считаем все префиксы затем берем и бинарным поиском ищем совпадение хешей.

#### Задание №3.

Условие:

3. \* Дана строка. Надо отвечать на запросы: является ли заданная подстрока палиндромом?  $O(\log n)$  на запрос.

Решение:

Посчитаем хеши с лева направо и справа налево для всех префиксов строки и перевернутой строки. Потом используя идею из первого задания и второго будем бинарным поиском идти и искать палиндромы относительно центра подстроки.

Задание №4.

Условие:

4. \* Дана строка. Надо отвечать на запросы: являются ли две заданные подстроки анаграммами?  $O(\log n)$  на запрос

Решение:

Задание №5.

Условие:

5. Посчитайте префикс функцию для строки, состоящей из вашего имени и фамилии.

Решение:

георгийхлучин

000010000000

Задание №6.

Условие:

6. Дана строка. Для каждого ее префикса посчитайте, сколько раз он встречается в строке как подстрока.  $O(n)$ .

Решение:

Обычная префикс функция

Задание №7.

Условие:

7. Даны две строки. Найдите их наибольшую общую подстроку.  $O(n \log n)$ .

Решение:

С помощью двоичного поиска найдем длину максимальной подстроки в двух строках. Т.е. мы сначала приравниваем  $L$  к длине наименьшей строки а потом бинарным поиском ищем ее в наибольшей строке уменьшая  $L$ . После того как мы найдем максимальную длину строки что сделаем не более чем за  $n_{min} * \log n_{max}$  мы сможем с помощью хеширования найти эту строку в другой.

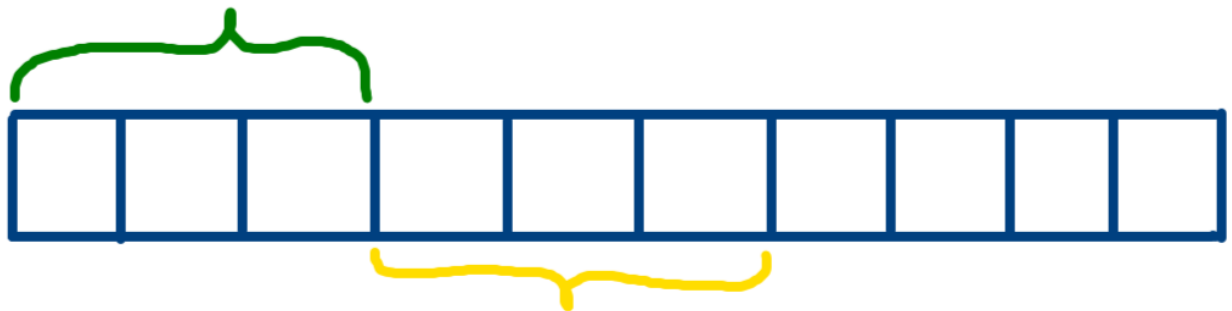
Задание №8.

Условие:

8. \* Дана строка. Найти все ее префиксы, которые являются палиндромами.  $O(n)$ .

Решение:

Сначала посчитаем все хеши для префиксов затем просто будем идти и сравнивать полученные хеши на отрезках.



Задание №9.

Условие:

9. Дана строка. Найти максимальную по длине строку, которая является ее префиксом, суффиксом, а так же встречается в середине строки.  $O(n)$ .

Решение:

Запускаем префикс функцию. И затем просто проходимся по полученному массиву находя наибольшие числа. В этих ячейках будут лежать концы наибольших подстрок.

Задание №10.

Условие:

10. Постройте строку с заданной Z-функцией.  $O(n)$ .

Решение:

Пусть в массиве  $Zz$  хранятся значения Z-функции, в  $Ss$  будет записан ответ. Пойдем по массиву  $Zz$  слева направо.

Нужно узнать значение  $S[i]s[i]$ . Для этого посмотрим на значение  $z[i]z[i]$ : если  $z[i]=0z[i]=0$ , тогда в  $S[i]s[i]$  запишем ещё не использованный символ или последний использованный символ алфавита, если мы уже использовали все символы. Если  $z[i] \neq 0z[i] \neq 0$ , то нам нужно записать префикс длины  $z[i]z[i]$  строки  $Ss$ . Но если при посимвольном записывании этого префикса в конец строки  $Ss$  мы нашли такой  $jj$  (индекс последнего символа строки), что  $z[j]z[j]$  больше, чем длина оставшейся незаписанной части префикса, то мы перестаём писать этот префикс и пишем префикс длиной  $z[j]z[j]$  строки  $Ss$ .

Для правильной работы алгоритма будем считать значение  $z[0]z[0]$  равным нулю. Заметим, что не всегда удастся восстановить строку с ограниченным алфавитом неподходящего размера. Например, для строки `abacabaabacaba` массив Z-функций будет `[0,0,1,0,3,0,1][0,0,1,0,3,0,1]`. Используя двоичный алфавит, мы получим строку `abababaabababa`, но её массив Z-функций отличается от исходного. Ошибка восстановления строки возникла, когда закончились новые символы алфавита.

Если строить строку по некорректному массиву значений Z-функции, то мы получим какую-то строку, но массив значений Z-функций от неё будет отличаться от исходного.

#### Задание №11.

Условие:

11. Постройте Z-функцию по префикс-функции, не восстанавливая строку.  $O(n)$ .

Решение:

Пусть в массиве  $Zz$  хранятся значения Z-функции, в  $Ss$  будет записан ответ. Пойдем по массиву  $Zz$  слева направо.

Нужно узнать значение  $S[i]s[i]$ . Для этого посмотрим на значение  $z[i]z[i]$ : если  $z[i]=0z[i]=0$ , тогда в  $S[i]s[i]$  запишем ещё не использованный символ или последний использованный символ алфавита, если мы уже использовали все символы. Если  $z[i] \neq 0z[i] \neq 0$ , то нам нужно записать префикс длины  $z[i]z[i]$  строки  $Ss$ . Но если при посимвольном записывании этого префикса в конец строки  $Ss$  мы нашли такой  $j$  (индекс последнего символа строки), что  $z[j]z[j]$  больше, чем длина оставшейся незаписанной части префикса, то мы перестаём писать этот префикс и пишем префикс длиной  $z[j]z[j]$  строки  $Ss$ .

Для правильной работы алгоритма будем считать значение  $z[0]z[0]$  равным нулю. Заметим, что не всегда удастся восстановить строку с ограниченным алфавитом неподходящего размера. Например, для строки `abacabaabacaba` массив Z-функций будет  $[0,0,1,0,3,0,1][0,0,1,0,3,0,1]$ . Используя двоичный алфавит, мы получим строку `abababaabababa`, но её массив Z-функций отличается от исходного. Ошибка восстановления строки возникла, когда закончились новые символы алфавита.

Если строить строку по некорректному массиву значений Z-функции, то мы получим какую-то строку, но массив значений Z-функций от неё будет отличаться от исходного.

#### Задание №12.

Условие:

12. \* Даны строки  $S$  и  $T$ , найдите в  $S$  подстроки, отличающиеся от  $T$  одной буквой.  
 $O(n)$

Решение:

Запустим префикс функцию, и счетчик который будем обновлять как только нашли одинаковое слово, т.е. счетчик идет до единицы если на отрезке длины слова содержится непрерывная часть слова то идем по ней префикс функцией

#### Задание №13.

Условие:

13. Дан набор строк  $s_i$ . Отсортировать их в лексикографическом порядке за время  $O(\sum |s_i|)$  (константный алфавит).

Решение:

Поразрядная сортировка

#### Задание №14.

Условие:

14. Дан набор строк  $s_i$ . Отвечать на запросы: по данным  $(i, j)$  найти длину наибольшего общего префикса строк  $s_i$  и  $s_j$ .

Решение:

Радикс сортом отсортируем все строки но будем хранить ссылки в начальном массиве. Когда будем искать наибольший общий префикс. То будем его искать только у слов находящийся в одних начальных ячейках. Хотя если использовать поразрядную сортировку по старшим индексам которая использует разбиение на группы, то можно запоминать и сравнивать именно ячейки.

Задание №15.

Условие:

15. Дан набор строк  $s_i$ . Найти самую короткую строку, которая не является префиксом никакой из них.

Решение:

Задание №16.

Условие:

16. . Отвечать на запросы: число вхождений строки  $s$  в  $t$ . Предподсчет за  $O(|t|)$ , ответ за  $O(|s|)$

Решение:

Обычная префикс функция.