SPb HSE, ПАДИИ, 1 курс, весна 2023/24
Практика по алгоритмам #16

Суффиксный массив

21 мая

Собрано 21 мая 2024 г. в 16:17

Содержание

Суффиксный массив

1. **Задачи про суффиксный массив**

   a) Наибольшая общая подстрока двух строк за $\mathcal{O}(|s| + |t|)$.
   b) Реализуйте сжатие `LZSS` за $\mathcal{O}(n \log n)$. А за $\mathcal{O}(n)$?

2. **ST → SA**

   По суффиксному дереву постройте суффиксный массив с LCP.

3. **SA → ST**

   По суффиксному массиву с LCP постройте суффиксное дерево.

4. **Словари offline**

   Даны словарь (конечное множество слов) и текст. Пример: $\{abcd, bc, a, da\}$ и текст $abcda$.

   a) Для каждого слова из словаря определить, входит ли оно как подстрока в текст.
   b) Для каждого слова из словаря найти число вхождений в текст.

5. **Поиск подматрицы**

   Даны матрицы чисел A и B. Проверить, является ли B подпрямоугольником A, $\mathcal{O}(|A|+|B|)$.

6. **Почти совпадения**

   Дан словарь. В онлайне поступают слова, нужно говорить «можно ли в данном слове заменить ровно одну букву, чтобы получить словарное слово».

7. `(*)` **РОИ-2004.**

   Даны n словарных слов и m слов текста. Суммарная длина всех $n+m$ слов равна L. Скажем, что слова похожи, если можно из каждого удалить не более одной буквы, чтобы они стали равны. Найдите для каждого слова текста:

   a) какое-нибудь похожее слово словаря;
   b) кол-во похожих слов в словаре.

Разбор задач практики

1. **Задачи про суффиксный массив**

   a) **Общая подстрока.** Строим суфмас $s\#t\#$ и считаем LCP. Для каждого суффикса s ищем ближайший слева и справа суффикс t, два указателя. Смотрим их LCP – минимум в очереди или Фарах-Колтон-Бендер.
   А можно смотреть только позиции, где рядом суффиксы из s и из t, все равно их LCP не меньше, чем у не соседних.

   b) **LZSS.** Пусть мы уже выписали i символов. Нужно быстро найти $j < i$: $\texttt{LCP}(i, j) = \max$. И жадно из i перейти в $i + \texttt{LCP}(i, j)$. Раньше мы перебирали все j за $\mathcal{O}(i)$. Теперь мы можем посмотреть на ближайший слева/справа в суффмассиве.
   Способ за $\mathcal{O}(n \log n)$. Будем держать позиции в суфмассиве всех $j < i$ в `set<int>` (нужны `insert`, `lower_bound`).
   Способ за $\mathcal{O}(n)$. Каждой позиции суфмассива соответствует начало суффикса $sa[i]$. Нужно найти ближайший справа и слева меньший элемент массива sa.
   Умеем это делать стеком за $\mathcal{O}(n)$ (см. 1-й семестр).
   Минимум `LCP` можно насчитывать, запоминая минимум между соседями на стеке. А можно написать Фараха-Колтона-Бендера.

2. **ST → SA**

   dfs по дереву, помним позицию самой высокой вершины с момента, как последний раз были в листе, эта высота и есть LCP.

3. **SA → ST**

   Пусть мы уже построили дерево для первых i суффиксов SA, стоим в листе. Поднимаемся от листа до уровня $lcp[i]$, создаём развилку, новый лист.

4. **Словари offline**

   Строим Ахо-Корасик для словаря.

   a) Проходим по тексту, помечаем все посещенные вершины. В конце проходим снизу вверх по бору и делаем `visited[suf[v]] |= visted[v]`.

   b) Проходим по тексту, считаем число посещений каждой вершины. В конце проходим снизу вверх по бору и делаем `count[suf[v]] += count[v]`.

5. **Поиск подматрицы**

   Хеш угла (r, c) матрицы $\sum_{ij} a_{ij} P^{r-i} Q^{c-j}$. Хеш подпрямоугольника – знакопеременная сумма.

6. **Почти совпадения**

   Сложим в хеш-таблицу все «словарные слова с одним пропуском». Пример `idea` → `*dea`, `i*ea`, `id*a`, `ide*`. При запросе, ищем «данное слово с одним пропуском в хеш-таблице». Для слова w за $\mathcal{O}(|w|)$ получаются хеши всех его версий с пропусками ⇒ решили за $\mathcal{O}$(размера входных данных).

7. (*) **РОИ-2004. Задача-ровесница.**

Для каждой фиксированной длины l словарного слова строим два бора всех словарных слов длины l — слова в прямом порядке, слова в обратном порядке. Онлайн отвечаем для слова текста — несколько запросов, как в (1)-й задаче практики. Проверить «есть ли точка в прямоугольнике». Кстати, посещавшие доплекции умеют это КД-деревом за $\mathcal{O}(\log n)$.

Домашнее задание

1. **(2) Суффиксный массив**

   Пусть у вас есть чёрный ящик «суффиксный массив + LCP строки s», работающий за $\mathcal{O}(|s|)$. Например, Каркайнен-Сандерс + Касаи.

   Задача: за $\mathcal{O}(|s|)$ найти самую длинную p, которая встречается в s минимум 3 раза. Найти не только длину p, но и саму строку p.

3.1. Дополнительная часть

1. **(3) Общая подстрока k строк**

   Предложите алгоритм поиска max общей подстроки k строк за их суммарную длину.

   Решите задачу суффиксным массивом, суффиксный массив умеют строить за $\mathcal{O}(n)$.

2. **(3) Количество строк**

   Дан словарь и число n.
   Посчитайте количество строк длины n над алфавитом $\{a, b\}$, которые не содержат ни одного словарного слова, как подстроку. Время $\mathcal{O}(nL)$, где L – суммарная длина слов в словаре.
   (+1) допбалл за $\mathcal{O}(L)$ памяти.