

SPb HSE, ПАДИИ, 1 курс, весна 2023/24
Практика по алгоритмам #15

Бор-2

14 мая

Собрано 14 мая 2024 г. в 17:27

Содержание

1. Бор-2	1
2. Разбор задач практики	2
3. Домашнее задание	4
3.1. Дополнительная часть	4

Бор-2

1. Словари offline

Даны словарь (конечное множество слов) и длинная строка t .

- Для каждого слова из словаря определить, входит ли оно как подстрока в t .
- Для каждого слова из словаря найти число вхождений в t .

$\mathcal{O}(|t| \cdot \max |word_i|)$. Мы уже умеем делать это хешами, сейчас используйте именно бор.

2. Сжатое суффиксное дерево

- Пусть мы хотим положить в бор все суффиксы строки s : $|s| = n$. Придумайте, как сохранить полученный бор, используя $\mathcal{O}(n)$ памяти.
- Полученная в предыдущем пункте структура – суффиксное дерево. Используя суффиксное дерево от текста, научитесь искать в online строки в тексте за $\mathcal{O}(|s|)$.

3. Разбиение на словарные слова

Дан словарь слов суммарной длины L и текст T . Слова длины $\leq l$.

- Представить T в виде конкатенации минимального числа словарных слов за $\mathcal{O}(L + l|T|)$. Слова можно использовать более одного раза.
- Представить T в виде конкатенации min числа **подстрок** словарных слов. $\mathcal{O}(L + |T|)$

4. Задачи про суффиксное дерево

- Найти количество различных подстрок.
- Найти самый длинный рефрен – подстроку s : $\text{count}(s) \cdot |s| \rightarrow \max$.
- Общая подстрока двух строк.

5. Бор – это не только про строки

Нужна структура данных, умеющая $\text{add}(x)$, $\text{del}(x)$, $\text{get-kth}(k)$ для целых 32-битных чисел.

6. XOR-1

Дан массив a из n целых чисел. За $\mathcal{O}(n)$ посчитайте $\forall a_i$ такое j : $a_i \oplus a_j$ максимально.

7. (*) XOR $\geq k$

Дан массив a из n целых чисел и число k . Посчитайте количество пар i, j : $a_i \oplus a_j \geq k$.

8. Поиск подматрицы

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

9. Почти совпадения

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

Разбор задач практики

1. Словари offline

Медленное решение. Строим бор. От каждой позиции текста «откладываем бор» — спускаемся по бору и отмечаем найденные слова.

2. Сжатое суффиксное дерево

- За $\mathcal{O}(n^2)$ просто кладем все суффиксы в бор.
- $\mathcal{O}(n)$ памяти (сжатое дерево): избавимся от вершин с одним ребенком. Теперь на ребре написана строка. Чтобы ребро занимало мало памяти, храним на нем $\langle \text{start}, \text{len} \rangle$. Тогда добавление строки в каком-то месте ответвляется и сразу создает новый лист, ≤ 2 новых вершин на каждый суффикс $\Rightarrow 2n$ вершин.

3. Разбиение на словарные слова

- Строим бор из словаря. $f[i]$ — минимальная стоимость выписать префикс длины i . Динамика вперёд: спускаемся по бору суффиксом $s[i:]$, если очередная вершина бора конечная, то $\text{relax}(f[i + \text{dep}], f[i] + 1)$. Максимальная глубина бора $l \Rightarrow$ время $\mathcal{O}(L + l|T|)$.
- Строим суффдерево для $s_1\#s_2\#\dots\#s_n$ за $\mathcal{O}(L)$. Жадно разбиваем текст за $\mathcal{O}(|T|)$: пока текст не пуст, отрезаем самый длинный префикс текста, по которому можем спуститься в боре.
Способ #2: Ахо-Корасиком. Он находит самую глубокую вершину бора, в которую можно прийти суффиксом текста. Считаем динамику, держим очередь с минимумом для окна, покрытого путем до текущей вершины бора.

4. Задачи про суффиксное дерево

В суффиксном дереве $s \forall x$ подстрока s заканчивается в вершине u или посреди ребра $v \rightarrow u$. В поддереве u заканчиваются все суффиксы начинающиеся в x (все вхождения x).

- Число подстрок.* В боре это число вершин. В сжатом боре это сумма длин ребер.
- Рефрен.* $\text{count}(s)$ — число суффиксов, кончающихся в поддереве, если спуститься по s . Заметим, что достаточно смотреть на строки, кончающиеся в вершинах.
- Строим суффиксное дерево от строки $s\#t$, считаем динамику «есть ли конец s -суффикса в поддереве» и «есть ли конец t -суффикса».

5. Бор – это не только про строки

Число = битовая строка длины 32 \Rightarrow задача решается бором, страшные биты ближе к корню.

6. XOR-1

Добавили все числа в бор, старшие биты ближе к корню. $\forall i$ ищем максимально отличающееся от a_i : пытаемся сперва спуститься по биту не равному биту a_i , если такого нет, штош.

7. (*) XOR $\geq k$

Построили бор от битовых строк. Старшие биты ближе к корню. Перебираем a_i . Спускаемся рекурсивно от корня в обе стороны. Если фиксированная часть пути от корня такова, что XOR уже точно меньше или точно больше k , выходим из рекурсии и `ans += (XOR < k ? 0 : size[vertex]);`. Т.е. мы не отсекаемся только для векти, которая даёт XOR ровно k .

8. Поиск подматрицы

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

9. Почти совпадения

Сложим в хеш-таблицу все «словарные слова с одним пропуском». Пример `idea` \rightarrow `*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 \Rightarrow решили за \mathcal{O} (размера входных данных).

Домашнее задание

1. (2) Отрезок словаря

Вам дан словарь из слов суммарной длины $L \leq 10^5$ над небольшим алфавитом размера k (например, $k = 26$). Постройте структуру данных, которая умеет за $\mathcal{O}(1)$ отвечать на запросы «сколько в словаре слов между S_1 и S_2 ».

2. (2) Online-запросы

Постройте структуру данных от текста длины не более 1000, которая умеет отвечать на запрос «найдите первое и последнее вхождение в текст строки s » за $\mathcal{O}(s)$.

3.1. Дополнительная часть

1. (2) XOR-3

Дан массив a длины n .

За время $\mathcal{O}(n)$ найдите отрезок массива с максимально возможным побитовым XOR.