

SPb HSE, ПАДИИ, 1 курс, весна 2023/24

Практика по алгоритмам #14

Бор и его друзья

30 апреля

Собрано 30 апреля 2024 г. в 14:41

Содержание

1. Бор и его друзья	1
2. Разбор задач практики	2
3. Домашнее задание	3

Бор и его друзья

1. Бор

Рассмотрим строки $S = \{s_1, s_2, \dots, s_n\}$ над алфавитом Σ .

Бором для S называется min корневое дерево, из каждой вершины которого по каждому символу Σ исходит не более одного ребра, и $\forall i s_i$ является корректным путём от корня.

- Как хранить рёбра бора? Минимизируйте время/память, максимизируйте удобство.
- Сделайте на основе бора `unordered_set<string>`.
- Сделайте на основе бора `unordered_map<string, int>`.
- Сделайте на основе бора `set<string>`.
- Что сделать, чтобы все строки заканчивались только в листьях?

2. Сортировка строк

Дан набор из n строк суммарной длины L над алфавитом A .

- Отсортируйте за время $\mathcal{O}(L \log |A|)$.
- Покажите, что \nexists алгоритма сортировки строк за $\mathcal{O}(L)$.
- (*) Отсортируйте за время $\mathcal{O}(L + |A|)$.

3. k -ая строка

Структура для строк: `add`, `del`, вернуть k -ую.

4. Т9

Дан словарь. Понимать быстро по строке s , сколько словарных слов на неё начинается.

А как вывести самое лучшее слово, начинающееся на s ?

А три самых лучших?

Как перестроить структуру, если вес (мера хорошеести) слова изменился?

Разбор задач практики

1. Бор

а) Хранение.

`v.next[c]` – ребро из вершины `v` по символу `c`.

`next` может быть массивом, `map`, `unordered_map`.

Если массив, то $\mathcal{O}(L|\Sigma|)$ памяти, иначе $\mathcal{O}(L)$, L – суммарная длина строк.

Подробнее в конспекте ПМИ за 3-й семестр.

б) `unordered_set<string>`. При добавлении строки создаем все нужные вершины и ребра, если их еще нет. В вершинах, где кончается строка, ставим пометку, что они конечные.

в) `unordered_map<string, int>`. Дополнительное поле в конечных вершинах.

г) `set<string>`. `v.next` – массив или `map`. Тогда можно перебирать строки в отсортированном порядке и делать `lower_bound`.

д) $s \rightarrow s\$$. Чтобы строки заканчивались только в листьях, добавим в конец каждой символ, не встречающийся в строках.

2. Сортировка строк

а) $\mathcal{O}(L \log |A|)$. Строим бор, в вершинах `map<char, int>`. `dfs` по бору, из вершины идем в детей в порядке возрастания символа.

б) Если очень большой алфавит и все строки длины 1, то сортировка строк не проще сортировки L чисел.

в) $\mathcal{O}(L + |A|)$. В вершине храним `unordered_map<char, int>`, добавляем строки. Чтобы в обходе дерева `dfs`-ом перебирать рёбра в порядке возрастания, нужно в каждой вершине отсортировать рёбра. Отсортируем все вершины сразу: сортируем поразрядно пары $\langle vertex, edgeChar \rangle$. Можно ещё чуть иначе? сортировать тройки $\langle j, s_i[j], i \rangle$, т.е. «позиция в строке, символ, номер строки», и класть в бор сразу упорядоченные ребра.

3. k -ая строка

Нужно поддерживать в боре «сколько строк заканчивается в поддереве», чтобы понимать по какому ребру спускаться к k -й.

4. Т9

Засунуть строки в бор, насчитать «сколько строк заканчивается в поддереве».

Ответ на задачу: спустить по строк, посмотреть, что хранится в вершине бора.

А ещё в каждом поддереве хранить «мах вес в поддереве».

Или, если просят, в каждом поддереве хранить сразу «три мах веса в поддереве».

Если вес поменялся, ок, пересчитаем максимумы на пути до корня.

Домашнее задание

1. (2) Т9

Дан набор n строк s_i . Для каждой s_i найдите min по длине префикс, который не является префиксом других строк. Время $\mathcal{O}(\sum |s_i|)$.

(+1), если у вас получится $\mathcal{O}(n)$ допамяти.