SPb HSE, ПАДИИ, 1 курс, осень 2024/25
Практика по алгоритмам #22

Корасик и Z-функция
6 марта

Собрано 5 марта 2025 г. в 22:18

Содержание

Корасик и Z-функция

1. **Z-функция**
   a) Найдите период строки с помощью Z-функции. $\mathcal{O}(n)$
   b) Найти кол-во различных подстрок. $\mathcal{O}(n^2)$. Только Z.
   c) Для строки узнать, какова её позиция в суффиксном массиве. $\mathcal{O}(n)$.
   d) Поиск с одной ошибкой. $\mathcal{O}(n)$.
   e) Поиск с двумя ошибками. $\mathcal{O}(n)$.

2. **Словари offline**

   Даны словарь (конечное множество слов) и текст.

   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.

3. **Словари online**

   Даны словарь (конечное множество слов) и текст. Обновлять ответ `online` при добавлении символа в конец текста.

   a) Пересчитать суммарное число вхождений слов из словаря в текст за $\mathcal{O}(1)$.
   b) Пересчитать множество всех вхождений слов из словаря в текст за $\mathcal{O}(1 + |\Delta A|)$, где ΔA – приращение ответа после добавления очередного символа.

4. **Манакер**

   Обобщите Z-функцию, чтобы искать число подпалиндромов строки за $\mathcal{O}(n)$.

5. **Хеширование множеств**

   Придумайте хеш-функцию, которая позволяет за $\mathcal{O}(1)$ делать 4 операции с множествами целых положительных 32-битных чисел.

   a) Добавить элемент в множество.
   b) Удалить элемент из множества.
   c) Проверить два множества на равенство.
   d) `+=` ко всем элементам множества.

6. **Один бор хорошо, а два – лучше!**

   Даны два бора A и B. Найдите для каждой вершины $u \in A$ самую глубокую вершину B, путь до которой равен суффиксу path: $root_A \rightsquigarrow u$. $\mathcal{O}(|A| + |B|)$. Размер алфавита $\mathcal{O}(1)$.

   Как увидеть в этой задаче обобщение Ахо-Корасик? Что есть A и B в Ахо-Корасик?

7. **Задачи про суффиксное дерево**

   a) Самая длинная подстрока, входящая в s дважды, причём вхождения не пересекаются.
   b) `(*)` Общая подстрока $k > 64$ строк за $\approx$ суммарную длину всех строк.

8. (*) ∞, **избегаемость шаблонов («вирусы»)**

   Дан словарь слов суммарной длины L. За время $\mathcal{O}(L)$ определите, существует ли бесконечная строка, не содержащая ни одно словарное слово как подстроку.

9. (*) **Хитрый поиск**

   Дан словарь, постройте структуру, чтобы быстро отвечать на запросы `get(s,t)` – число строк в словаре, которые начинаются с s, заканчиваются на t.

10. (*) **Почти совпадения**

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

Разбор задач практики

1. **Z-функция**
   a) Проверка для периода t: $z[t] = n-t$.
   b) $\forall i$ насчитаем z от суффикса $s[i{:}n)$ и прибавим к ответу $n - i - \max z_j$, т.е. те префиксы i-го суффикса, которые не встречаются правее.
   c) Знаем LCP. Сравнить s с суффиксом на больше/меньше можно за $\mathcal{O}(1)$.
   d) Поиск с одной ошибкой. $\mathcal{O}(n)$.

      Ищем s в t. Переберём начало вхождения i. Чтобы проверить i за $\mathcal{O}(1)$,
      хотим узнать LCP суффикса $t[i:]$ и строки s, это ровно $z(s\#t)[i + |s| + 1]$ (z-функция).
      После LCP совпадений идёт или конец строки, или ошибка. Осталось проверить равенство куска строки после ошибки: или хеши, или посмотреть на $z(\overline{s}\#\overline{t})$.
   e) Поиск с двумя ошибками. $\mathcal{O}(n)$.

      Переберем позицию i начала вхождения s в t.
      С помощью $z(s\#t)$ ищем максимальный общий префикс s и $t[i{:}i{+}|s|)$, после него позиция ошибки. С помощью $z(\overline{s}\#\overline{t})$ ищем максимальный общий суффикс s и $t[i{:}i{+}|s|)$, перед ним позиция ошибки. Сравниваем середину хешами.

2. **Словари offline**

   Строим автомат Ахо-Корасик для словаря. Затем в самом конце:
   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]`.

3. **Словари online**

   Строим Ахо-Корасик для словаря.
   a) Посчитаем динамику: количество терминальных вершин на суффиксном пути.
   b) Посчитаем супер-суффиксные ссылки: ближайшая терминальная вершина на суффиксном пути.

4. **Манакер**

   Ищем $R[i]$ – «радиус» палиндрома с центром i (округленную вверх половину длины).
   Пусть правее всех найденных кончается палиндром с центром i. Назовем его «текущий палиндром», он аналогичен текущим границам в вычислении Z-функции.
   Ищем $R[j]$, изначально полагаем $R[j] = \min(R[i - (j - i)], i + R[i] - j)$.
   Дальше в лоб расширяем $R[j]$. Расширится, только если перевалили за границу текущего палиндрома. Тогда текущим станет палиндром с центром j. Время линейно потому, что каждое успешное сравнение двигает правую границу текущего палиндрома.

   Чётные можно считать отдельно, а можно найти только нечётные в строке $s_1\#s_2\#\ldots\#s_n$.

5. **Хеширование множеств**

   Мы можем поддерживать для множества A величину $\sum_{a\in A} f(a) \bmod m$.

Осталось придумать f: легко посчитать и сложно подделать. $f(a) = P^a \bmod m$.
$\forall$ `a a+=Δa` $\Leftrightarrow$ $f(a)$`*=`$P^{\Delta a} \bmod m \Rightarrow$ все три операции `add`, `del`, `+=` за $\mathcal{O}(\log a)$.
Вероятность ошибки: P случайное и должно быть корнем многочлена $\deg = \max a \Rightarrow$ $\Pr \leqslant \frac{\max a}{m}$. $\max a \leqslant 10^9, m \leqslant 10^{18} \Rightarrow$ `OK`. Но для $A_1 = \{0\}$, $A_2 = \{m-1\}$ хеши совпадут $\forall P$.

Мемное решение. Пусть $f(x) =$ `mem`$[x]$, где `mem` — хеш-таблица, куда при первом обращении кладётся псевдорандом. $\Pr$[ошибки] $= \frac{1}{m}$, время `add`, `del` $\mathcal{O}(1)$, но много лишней памяти.

Третий вариант. $A = \prod_{a \in A}(z-a) \bmod m$, где z фиксированное случайное.
Множества A и B имеют коллизию $\Leftrightarrow$ z — корень многочлена степени $\leqslant \max |A|, |B| \Rightarrow$ $\Pr$[ошибки] $\leqslant \frac{|A|}{m}$. `add` за $\mathcal{O}(1)$, `del` за $\mathcal{O}(\log m)$ т.к. нужно обращать по модулю.

Итого. Первое — единственное решение с `+=`. Хеш-таблица даёт самое быстрое решение, но ест много памяти. Третье решение лучше первого по ошибке и времени `add`. В нашей паре нужно именно *первое решение*.

6. **Один бор хорошо, а два – лучше!**

   Замкнем бор B до полного автомата за $\mathcal{O}(|B|)$. Для каждой вершины $u \in A$ ответ пересчитываем через предка: `ans[a[v][c]] = b[ans[v]][c]`.

7. **Задачи про суффиксное дерево**

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

   a) *Дважды входящая.* Ищем v: $\min(R[v]-L[v], \text{depth}[v]) \to \max$. Ответ может быть посреди ребра.
   b) `(*)` *Общая подстрока* $\forall k$. За $\mathcal{O}(n)$ посчитаем для каждой вершины количество различных чисел в поддереве. Было в прошлом семестре через LCA и суммирование снизу вверх по дереву.

8. `(*)` ∞**, избегаемость шаблонов («вирусы»)**

   Строим автомат Ахо-Корасик для словаря. Запретим (выкинем из автомата) вершины, на пути из суффссылок от которых есть запрещённые слова. Проверим, есть ли в графе цикл, достижимые из стартовой вершины. Бесконечная строка $\exists$ iff $\exists$ такой цикл.

9. `(*)` **Хитрый поиск**

   `get(s,t)` – бор прямых строк, отдельно бор обратных строк. Конечные вершины помечены номерами строк.
   Спускаемся в первом боре по s, во втором по $\overleftarrow{t}$, получили два поддерева. Ответ – пересечение множеств пометок в этих поддеревьях. В каждом дереве все пометки различны, умеем решать такую задачу 2D-запросом.

10. `(*)` **Почти совпадения**

    Сложим в хеш-таблицу все «словарные слова с одним пропуском». Пример `idea` $\to$ `*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) Первое и последнее вхождение**

   Даны словарь и текст. Найдите для каждого словарного слова первое и последнее его вхождения в текст в качестве подстроки.

2. **(2) Навигация в боре**

   Даны бор A и строка s. Найдите вершину бора v, от которой строку s можно отложить вниз по бору. Размер алфавита $\mathcal{O}(1)$. Время $\mathcal{O}(|A| + |s|)$.

3.1. Дополнительная часть

1. **(2) 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.

   (+0.5) за $\mathcal{O}(L)$ памяти.

2. **(2) Быстрый Ахо-Корасик**

   Дан бор A над алфавитом S произвольного размера, постройте суффиксные ссылки за $\mathcal{O}(|A| \cdot poly(\log |S|))$. На самом деле за такое время можно построить полный автомат.

   Мы уже умеем строить суффиксные ссылки несколькими способами.
   Полный автомат мы строим за $\mathcal{O}(|A| \cdot |S|)$. `bfs` + откаты, как в префикс функции, работают за $\mathcal{O}(\Sigma|s_i|)$, что может быть сильно больше $|A|$.

Разбор домашнего задания

4.1. Обязательная часть

1. ?

4.2. Дополнительная часть

1. ?