SPb HSE, ПАДИИ, 1 курс, весна 2023/24
Практика по алгоритмам #12

Полиномиальные хеши
16 апреля

Собрано 16 апреля 2024 г. в 11:08

Содержание

Полиномиальные хеши

0. Чёрный ящик «хеши»: умеем предподсчёт за $\mathcal{O}(n)$ и $\forall$ подстроки $[l, r]$ за $\mathcal{O}(1)$ возвращать её хеш. Если хеши строк совпадают, считаем строки равными.

   Упражнение: поиск подстроки в строке с помощью хешей за $\mathcal{O}(n + m)$.

1. Минимальный период строки с помощью хешей.
2. Число различных подстрок за $\mathcal{O}(n^2)$.
3. Наибольшая общая подстрока двух строк за $\mathcal{O}(n \log n)$.
4. Поиск по словарю. Есть длинный текст и словарь коротких слов (длина $\leqslant 30$). Нужно для каждого слова определить, сколько раз оно встречается в тексте.
5. Научиться сравнивать любые две подстроки на больше-меньше.
6. Построить суффиксный массив за $\mathcal{O}(n \log^2 n)$ времени, $\mathcal{O}(n)$ памяти.
7. С помощью суффиксного массива текста научиться в online находить вхождение s за $\tilde{\mathcal{O}}(|s|)$.
8. Научитесь ещё раз решать задачу про словарь. Теперь суффиксным массивом.
9. (*) Найдите самый длинный палиндром строки за $\mathcal{O}(n \log n)$. За $\mathcal{O}(n)$.
10. (*) Жил был геном – циклическая ACGT-строка длины 10^6. Вам дают его 10^6 случайных подстрок длины 100 каждая. Нужно восстановить геном.

Разбор задач практики

0. **Чёрный ящик**

   Алгоритм Рабина-Карпа. См. конспект.

1. **Период с помощью хешей.**

   Чтобы проверить, период ли префикс длины t, сравним $s[0{:}n-t)$ и $s[t{:}n)$.

2. **Число различных подстрок за $\mathcal{O}(n^2)$.**

   Добавим все хеши в хеш-таблицу.

3. **Наибольшая общая подстрока двух строк за $\mathcal{O}(n \log n)$.**

   Бинпоиск по ответу. Внутри хеши всех подстрок длины k первой строки кладём в хеш-таблицу, а хеши подстрок длины k второй строки там ищем.

4. **Поиск по словарю.**

   Возьмём все $30 \cdot |text|$ хешей подстрок текста и положим в хеш-таблицу.

5. **Научиться сравнивать любые две подстроки на больше-меньше.**

   Бинпоиск по длину «совпавшей части».

6. **Построить суффиксный массив за $\mathcal{O}(n \log^2 n)$ времени, $\mathcal{O}(n)$ памяти.**

   Суффикс задаётся позицией начала $\Rightarrow$ суфмассив = перестановка чисел.

   Стандартному `sort` передаём компаратор из предыдущей задачи.

7. **Научиться в online находить вхождение s за $\tilde{\mathcal{O}}(|s|)$.**

   Бинпоиск по суффиксному массиву. Внутри за линию сравниваем. $\mathcal{O}(|s| \log |text|)$.

8. **Ещё раз решать задачу про словарь. Теперь суффиксным массивом.**

   Кол-во вхождений = первое и последнее вхождения = два бинпоиска.

9. **(*) Самый длинный палиндром строки за $\mathcal{O}(n \log n)$.**

   Палиндром задаётся позицией центра. Радиус можно искать бинпоиском. Нужны хещи и прямой и перевёрнутой строки.

10. **(*) Жил был геном**

    Жадность: соединять две строки с самым длинным зацеплением. Чтобы находить самое длинное зацепление, будем класть хеши подстрок длины k в хеш-таблицу. По убыванию k.

Домашнее задание

1. **(2) Общая подстрока k строк**

   Предложите алгоритм поиска наибольшей общей подстроки k строк длины n.
   Оцените и время работы, и используемую память.

2. **(2) Разбить строку на палиндромы**

   Дана строка s, представьте её в виде конкатенации минимального числа палиндромов.
   Нужно решение за $\mathcal{O}(|s|^2)$ времени и $\mathcal{O}(|s|)$ памяти.

3.1. Дополнительная часть

1. **(2) Поиск с k ошибками**

   Найти подстроку в тексте. При сравнении строк, если несовпадений символов было не более k, строки считаются равными. $\mathcal{O}(nk \log n)$.