Полиномиальные Хэш-функции

Простая в реализации альтернатива алгоритмам на строках и матрицах

Определение

- Для данного одномерного массива целых чисел *S* длины *l*, некоторого положительного числа *x* и некоторого положительного числа *m* полиномиальной хеш-функцией *p*(*S*,*x*,*m*) называется

$$p(S,x,m) = \left(\sum_{i=0}^{l-1} S_i x^i\right) \bmod m$$

Рекурентное Определение

- Для данного одномерного массива целых чисел *S* длины *l*, некоторого положительного числа *x* и некоторого положительного числа *m* полиномиальной хеш-функцией *p*(*S*,*x*,*m*) называется

$$p(S, x, m) = \left(\sum_{i=0}^{l-1} S_i x^i\right) \bmod m$$

Основное Свойство

- Для данного одномерного массива целых чисел *S* длины *l*, некоторого положительного числа *x* и некоторого положительного числа *m* полиномиальной хеш-функцией *p*(*S*,*x*,*m*) называется

$$p(S,x,m) = \left(\sum_{i=0}^{l-1} S_i x^i\right) \bmod m$$

Пример

- Для данного одномерного массива целых чисел *S* длины *l*, некоторого положительного числа *x* и некоторого положительного числа *m* полиномиальной хеш-функцией *p*(*S*,*x*,*m*) называется

$$p(S,x,m) = \left(\sum_{i=0}^{l-1} S_i x^i\right) \bmod m$$

Поиск подстроки в строке 1/2

- Для данного одномерного массива целых чисел *S* длины *l*, некоторого положительного числа *x* и некоторого положительного числа *m* полиномиальной хеш-функцией *p*(*S*,*x*,*m*) называется

$$p(S, x, m) = \left(\sum_{i=0}^{l-1} S_i x^i\right) \bmod m$$

Поиск подстроки в строке 2/2

- В худшем случае время $O(sr)$.
- При случайном x и большом *mod* при совпадении хешей можно опустить посимвольную проверку.
- Для простоты вместо взятия модуля можно доверить переполнению *unsigned long long*.
- Тогда в худшем случае время: $O(s+r)$ и память $O(s+r)$.
- В точности равно сложности КМП.

Префикс-функция 1/2

- Для данного одномерного массива целых чисел *S* длины *l*, некоторого положительного числа *x* и некоторого положительного числа *m* полиномиальной хеш-функцией *p*(*S*,*x*,*m*) называется

$$p(S,x,m) = \left(\sum_{i=0}^{l-1} S_i x^i\right) \bmod m$$

Префикс-функция 2/2

- Сложность: $O(r + s \log r)$.
- Сложность КМП: $O(r + s)$.
- КМП заметно проще в реализации.

Z-функция

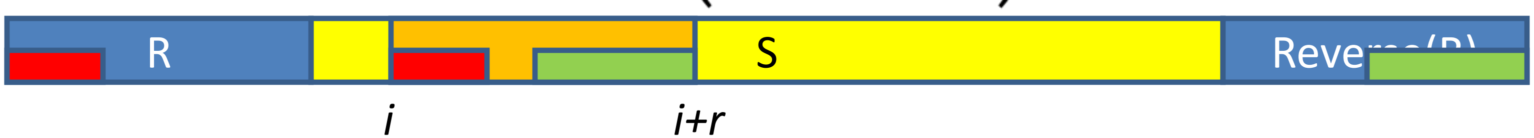
- Для данного одномерного массива целых чисел *S* длины *l*, некоторого положительного числа *x* и некоторого положительного числа *m* полиномиальной хеш-функцией *p*(*S*,*x*,*m*) называется

$$p(S,x,m) = \left(\sum_{i=0}^{l-1} S_i x^i\right) \bmod m$$

Неточный поиск подстроки 1/3

- Для данного одномерного массива целых чисел *S* длины *l*, некоторого положительного числа *x* и некоторого положительного числа *m* полиномиальной хеш-функцией *p*(*S*,*x*,*m*) называется

$$p(S,x,m) = \left(\sum_{i=0}^{l-1} S_i x^i\right) \operatorname{mod} m$$



Неточный поиск подстроки 2/3

- Задача: для строк S и R найти вхождения в S всех строк, отличающихся от R не более, чем *K* символами.
- Решение: пусть $Q=substr(S, i, i + r)$, P=R
  - Найдем длину *l1* максимального общего префикса Q и P. Очевидно, l1+1-ый символ различен.
  - Удалим первые l1+1 символов из обеих строк.
  - Повторим предыдущие два шага K раз, или пока P не станет равно R.
  - Если после *j* итераций строки равны, они имеют ровно *j* различных символов.

Неточный поиск подстроки 3/3

- Сложность: $O(r + s\ K \log r)$
- Существует алгоритм за $O(r + s\ K)$ или быстрее?

Поиск минимального лексикографического сдвига строки

- Задача: для заданной строки S, среди всех ее циклических сдвигов, найти лексикографически минимальный.
- Решение:
  - Приписать строку к самой себе;
  - Запомнить первую позицию как текущего кандидата;
  - Для каждой позиции начиная со второй улучшить кандидата, сравнив первый различный символ.
- Сложность: $O(s \log s)$
- Сложность алгоритма Дюваля: $O(s)$

Сортировка циклических сдвигов строки

- Задача: для данной строки S длины s отсортировать лексикографически все ее циклические сдвиги.
- Решение:
  - Приписываем S к самой себе.
  - Сортируем массив $[0..s\text{-}1)$.
  - Компаратор: первый различный символ двух сдвигов. Сложность компаратора: $O(\log s)$.
- Сложность алгоритма: $O(s \log s \log s)$.
- Сложность суффиксных массивов: $O(s \log s)$

Полиномиальные хеши в матрицах

- Для данного одномерного массива целых чисел *S* длины *l*, некоторого положительного числа *x* и некоторого положительного числа *m* полиномиальной хеш-функцией *p*(*S*,*x*,*m*) называется

$$p(S, x, m) = \left(\sum_{i=0}^{l-1} S_i x^i \right) \bmod m$$

Поиск подматрицы в матрице

- Для данного одномерного массива целых чисел *S* длины *l*, некоторого положительного числа *x* и некоторого положительного числа *m* полиномиальной хеш-функцией *p*(*S*,*x*,*m*) называется

$$p(S, x, m) = \left(\sum_{i=0}^{l-1} S_i x^i\right) \bmod m$$

Вывод

- Один простой алгоритм вместо десятка сложных;
- Для всех задач сложность не более чем в $\log(n)$ раз хуже лучшего известного алгоритма.
- Некоторые задачи не имеют асимптотически приемлемого решения без полиномиальных хешей.