**Полиномиальное хеширование строк**

Основная идея хеширования (использования хеш-функций) следующая. Каждому объекту сопоставим некоторое число, которое называется *хешем*. Сама функция, которая для данного объекта вычисляет это число, называется *хеш-функцией*.

Хеш-функция должна обладать следующими свойствами:

1. Она должна относительно легко вычисляться для объекта.
2. Значения функции должны быть "похожи" на случайные, т.е. значения хеш-функции должны быть мало предсказуемы.
3. Значения хеш-функции должны быть равномерно (или близко к равномерному) распределены, не должно быть значений хеш-функции, которое встречается существенно чаще других значений.
4. Необходимо тем самым минимизировать вероятность коллизии, то есть совпадения значений хеш-функции для двух различных объектов, для достижения этого как раз и требуется близость распределения значения хеш-функции к равномерному.

Хеширование часто применяется для решения задач на строки.

Для начала рассмотрим алгоритм вычисления хеш-функции для строк.

Определим значение хеш-функции для строк длины 1. Каждому возможному символу строки сопопоставим уникальное целое положительное число~-— значение хеш-функции от этого символа. Например, можно взять ASCII-код или Unicode символа или если строка состоит только из строчных латинских букв, можно взять h(′a′)=1, h(′b′)=2, …, h(′z′)=26. Пустой строке будет соответствовать нулевое значение хеш-функции, поэтому не следует использовать число 0 для значения хеш-функции от какого-либо одного символа.

Для строк длины 2 желательно выбрать такой алгоритм вычисления хеш-функции, который  
обладал бы следующими свойствами:

1. Значение хеш-функции для всех строк длины 1 и 2 были бы различными.
2. Использовались бы значения хеш-функции для одного символа, определенные ранее.
3. Алгоритм легко обобщается на строки большей длины.

Простое решение — для строки из двух символов a0a1 значение хеш-функции вычислить по формуле h(a0a1)=h(a0)b+h(a1). Если при этом выбрать значение b большим, чем максимальное возможное значение хеш-функции от одного символа (например, если используются только латинские буквы, то можно взять b=27), то тогда у всех строк длины 2 значения хеш-функции будут различными.  
  
Для строк длины 3 можно определить значение хеш-функции следующим образом: h(a0a1a2)=h(a0)b2+h(a1)b+h(a2) и т.д. Например, если взять b=27, то хеш-функция от строки 'one' будет равна h(′one′)=h(′o′)b2+h(′n′)b+h(′e′)=15⋅272+14⋅27+5=11318.  
  
Соответственно, для строки длины n значение хеш-функции можно определить так: h(a0a1a2…an−1)=h(a0)bn−1+h(a1)bn−2+…+h(an−2)b+h(an−1). Это значение также равно значению числа a0a1a2…an−1¯ в системе счисления с основанием b, то есть числом, записанного символами a0a1a2…an−1,  если считать, что один символ соответствует одной цифре в системе счисления с основанием b, а значением цифры является значение хеш-функции от этого символа. Также можно рассмотреть это, как значение многочлена степени n−1 с коэффициентами a0, a1, …, an−1, вычисленное при x=b.  
  
Но при длинной строке результатом вычисления по этой формуле будет очень большое число (длина которого будет порядка длины строки), поэтому использовать такой хеш для решения задач нельзя (так как основным смыслом хеш-функции явлется сопоставление большому объекту некоторого относительно небольшого числа), поэтому в качестве значения хеш-функции возьмем остаток от деления полученного числа на некоторое большое число M. В качестве значения M рекомендуется выбирать большое простое число, также часто в качестве значения M выбирают степень числа 2, равную размеру целочисленного типа данных, например, 232 или 264. Обязательное требование к числу M~-— оно должно быть взаимно простым с числом b. Итак, определим значение хеш-функции от строки:

h(a0a1a2…an−1)=(h(a0)bn−1+h(a1)bn−2+…+h(an−2)b+h(an−1))modM

Значение определенной таким образом хеш-функции удобно вычислять, как и значение многочлена по схеме Горнера пользуясь свойством:

h(a0a1a2…an−1)=

=((h(a0)bn−2+h(a1)bn−3+…+h(an−2))b+h(an−1))modM=

=(h(a0a1a2…an−2)b+h(an−1))modM

То есть чтобы вычислить хеш-функцию от строки a0a1a2…an−1 необходимо взять  
вычислить хеш-функцию от строки без одного последнего символа a0a1a2…an−2,  
умножить результат на b, добавить значение последнего символа an−1 и взять  
остаток от деления результата на M. Например, это можно сделать при помощи следующей функции за время, пропорциональное длине строки:

**const int BASE = 27;**

**const int M = 1000000007;**

**int get\_hash(string & s)**

**{**

**int result = 0;**

**for (auto c: S)**

**result = ((long long)result \* BASE + c - 'a' + 1) % M;**

**return result**

**}**

Если же в конец данной строки S добавить строку T длиной в m символов, то число, соответствующее хеш-функции строки S необходимо сдвинуть на m символов влево, что соответствует умножению на bm, после чего добавить хеш-функцию строки T.

h(ST)=(h(S)bm+h(T))modM

То есть легко вычислить хеш-функцию от конкатенации двух строк S и T, если известны хеш-функции от данных строк.  
  
С другой стороны, пусть известна хеш-функция h(ST) и хеш-функция от ее префикса h(S). Тогда можно вычислить хеш-функцию от строки T:

h(T)=(h(ST)−h(S)bm)modM

Таким образом, если дана строка S и необходимо вычислить хеш-функцию от подстроки этой строки S[i..j] (то есть от подстроки из символов от i до j длиной j−i+1), то это можно сделать, зная значения хеш-функции от префиксов строки длины i−1 и j:

h(S[i..j])=(h(S[1..j])−h(S[1..j])bj−i+1)modM

То есть если известны значения хеш-функции на всех префиксах данной строки S, то значение хеш-функции от любой ее подстроки можно вычислить за O(1), если также известно значение bmmodM. Это можно сделать при помощи следующей функции:

**int get\_hash(int i, int j, int \* prefix, int \* power) {**

**return (prefix[j] - (long long)prefix[i-1] \* power[j-i+1] % M + M) % M;**

**}**

Эта функция получает на вход следующие параметры: границы подстроки i и j и два массива prefix и power. Сама строка S не передается в функцию, вместо нее передаются указанные массивы. В массиве prefix хранятся значения хеш-функции для всех префиксов строки S, то есть значение prefix[i] равно значению хеш-функции для префикcа S[1..i] строки S. В списке [ower хранятся значения степеней числа b по модулю M, то есть power[i]=bimodM.

В языке C++ удобно считать, что символы строки нумеруются начиная с 1, то есть строка состоит из символов s1s2…sn, тогда prefix[0]=0 и соответствует хеш-функции пустой строки.  
  
Значения массивов  prefix и power необходимо вычислить один раз в начале, выполнив так называемый ``предподсчет'', то есть предварительную подготовку, необходимую для реализации алгоритма. Для выполнения предподсчета используются две функции: функция calc\_prefix получает на вход строку S и заполняет массив prefix, содержащий значение хеш-функции для всех префиксов строки S, и функция calc\_power, получающая на вход число n и заполняющая массив power.

**const int BASE = 27;**

**const int M = 1000000007;**

**void calc\_prefix(string & s, int \* prefix)**

**{**

**int n = s.size();**

**prefix[0] = 0;**

**for (int i = 0; i < n; ++i)**

**prefix[i+1] = ((long long)prefix[i] \* BASE + s[i] - 'a' + 1) % M;**

**}**

**void calc\_power(int n, int \* power)**

**{**

**power[0] = 1;**

**for (int i = 0; i < n; ++i)**

**power[i+1] = (long long) power[i] \* BASE % M;**

**}**

Предподсчет выполняется за время O(n), но после однократно выполненного предподсчета  
функция get\_hash позволяет за O(1) вычислять значение хеш-функции для любой подстроки  
данной строки, что позволяет, например, быстро проверять на равенство две подстроки.