**БИОАЛГОРИТМИКА**

**15.08 Занятие 6**

1. **Хэширование**

В алгоритме поиска длинных общих фрагментов мы столкнулись вот с чем. Пусть N – длины сравниваемых последовательностей (для простоты считаем, что они примерно равны); k – минимальная длина нужного нам сходства, причем 4k >> N. Пример: N = 106; k = 20. Тогда время на построение таблицы вхождений O(4k) будет значительно больше времени работы основной части алгоритма.

Но: общее количество разных пятерок в последовательности длины N – примерно N (почему?). Поэтому при 4k >> N большинство строк в таблице вхождений будут пустыми.

Поступим следующим образом. (1) Создадим *небольшую* таблицу – размером T ~ 2N или 3N, что все еще много меньше, чем 4k; (2) выберем функцию *h: [1..4k] -> [1..T]* (то есть отображающую числа из множества *[1..4k]* в числа из множества *[1..T]*); (3) начальные позиции k-слова номер *x* будем записывать в строку под номером *h(x).*

Замечание 1. k-слово – это короткое выражения для «слово длины k». Каждое k-слово можно рассматривать как запись числа в системе счисления с основанием *k.* То есть, все k-слова можно считать перенумерованными от 0 до 4k-1. Мы по привычке будем считать, что k-слова перенумерованы номерами от 1 до 4k (просто прибавим к старому номеру 1).

Замечание 2. Так как у нас всего будет не более *N* разных *k-*слов, а в таблице ~3N строк, то *обычно* конкретной строке *z* будет соответствовать только одно *k-*слово! Хотя чисел *x* таких, что *h(x) = z* много – в среднем 4k */* T.

Замечание 3. Функция *h* называется хэш-функцией, а таблица из T строк – хэш-таблицей. А процедура построения хэш-таблицы называется хэшированием (посмотрите в словаре, что означает английский глагол *to hash*).

Пример хэш-функции для k-слов. Выбираем число *s<k* так, чтобы 4s ~ 3N. В качестве хэш-значения для k-слова *v* берем номер его начального фрагмента (префикса) длины s.

Более сложный, но часто используемый на практике пример хэш функции. (Это набросок описания. Если непонятно – не страшно. Поэтому он дан более мелким шрифтом). Выбираем число *b* (обычно ~ 20) и число *p* (обычно простое или степень двойки) так, чтобы было *p~ 3N*. Рассматриваем *k-*слово, как запись числа в системе счисления по основанию *b.* В качестве хэш-значения берем остаток от деления этого числа на *p.*

1. **Поиск неточных сходств.**

См. презентацию.