

Лекция 19. Хеширование

Курс «Программирование»

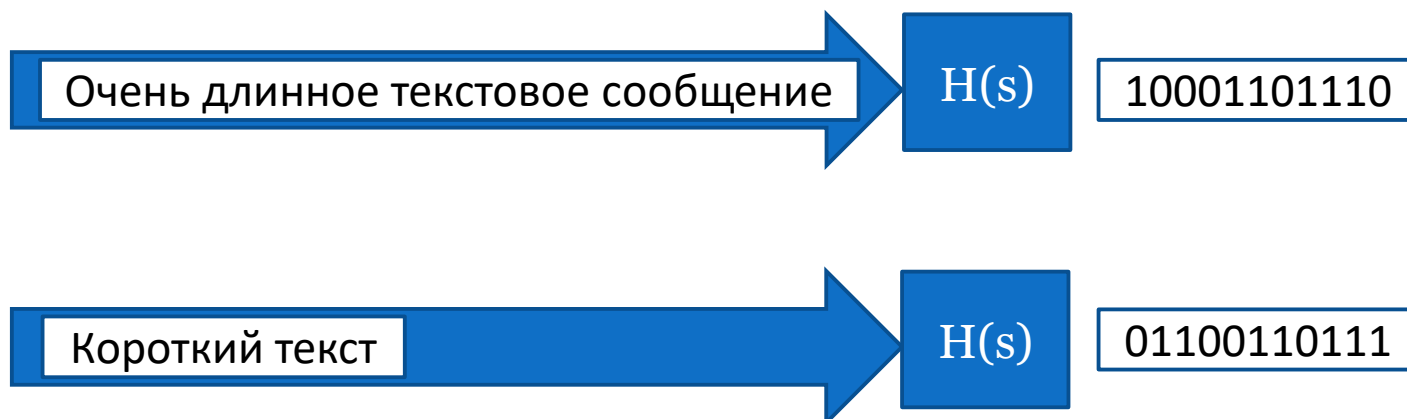
КИТ, 2 семестр

Щукин Александр

Валентинович

Хеширование (hashing)

1. Преобразование входного массива данных произвольной длины в выходную битовую строку фиксированной длины, выполняемое определенным алгоритмом
2. Алгоритм преобразования – хеш-функция (функция свертки)
3. Способ организации данных – хеш-таблица
4. Результат преобразования – хеш-код, хеш



Применение

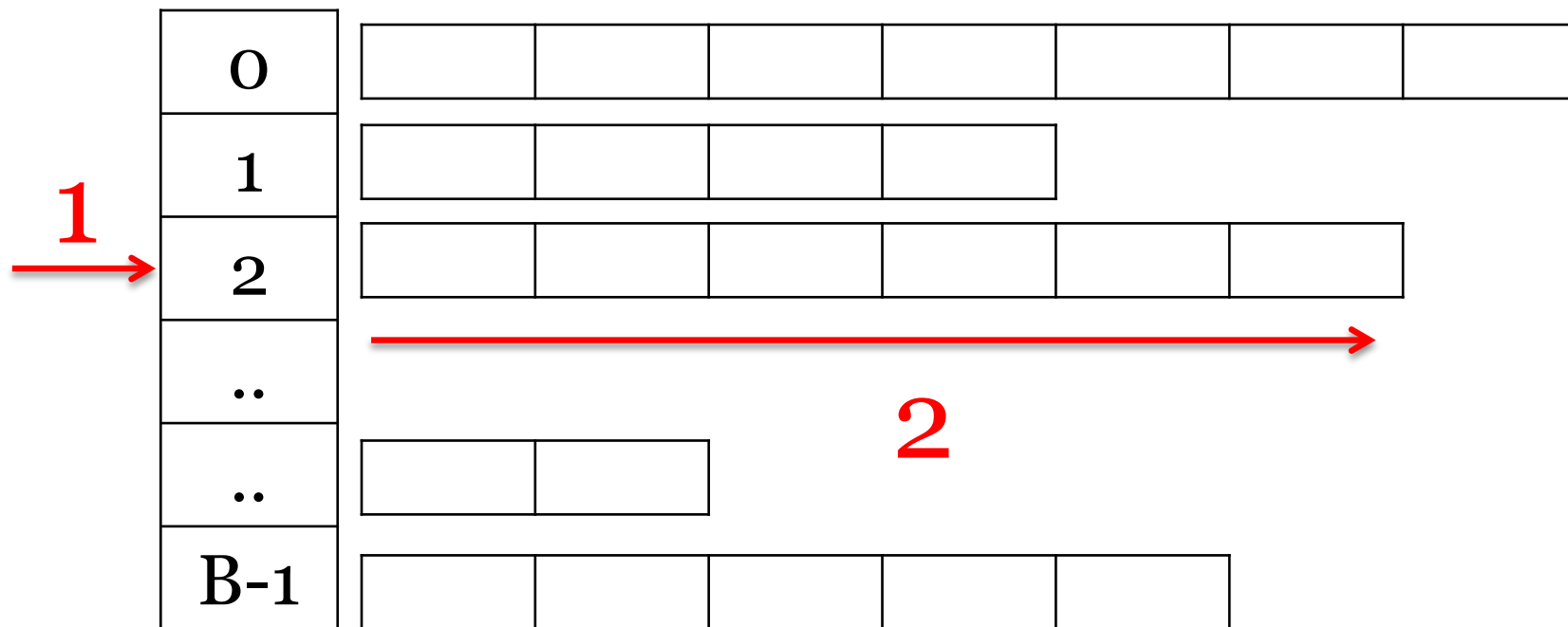
- Организация данных, основанных на множествах: ассоциативные массивы, словари
 - Алгоритмы поиска
 - Вычисление контрольных сумм
 - Криптография
 - Блокчейн
-
- Метод хеширования требует фиксированного времени на выполнение операторов с элементами множества (массива например).
 - В худшем случае время выполнения $O(N)$, но может быть быстрее

Открытое хеширование

0							
1							
2							
..							
..							
B-1							

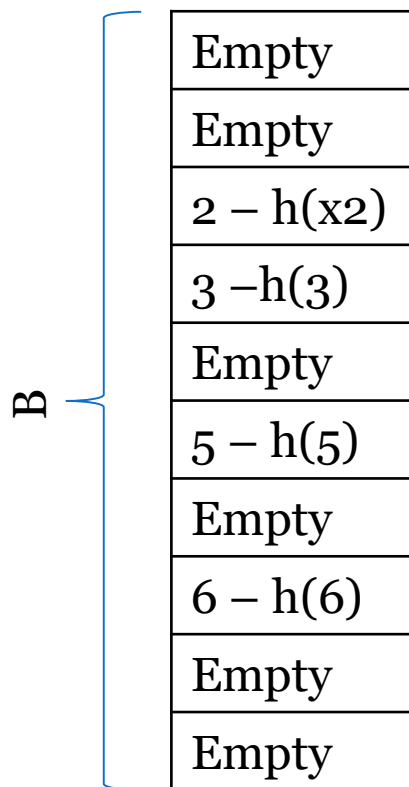
- Множество T разбивается на B классов (сегментов), от 0 до $B-1$
- Для каждого элемента x множества T хеш-функция $h(x)$ принимает целочисленное значение от 0 до $B-1$
- x – ключ. Хеш-код – $h(x)$

Алгоритм поиска



- $O(1 + N/B)$ – оценка эффективности

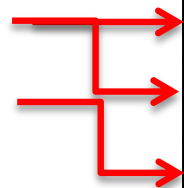
Закрытое хеширование



Empty
Empty
2 – $h(x_2)$
3 – $h(3)$
Empty
5 – $h(5)$
Empty
6 – $h(6)$
Empty
Empty

- В саму ячейку записывается значение x из множества T
- Номер ячейки определяется хеш-кодом $h(x)$. Изменяется в пределах $[0, B-1]$
- Если мощность множества T больше B , то возникает эффект коллизии
- Реструктуризация:
 - Для закрытого хеширования при $N \geq 0.9B$
 - Для открытого хеширования при $N \geq 2B$ или $N \geq 3B$

Решение коллизии



Empty
Empty
2 – h(x2)
3 – h(3)
Empty
5 – h(5)
Empty
6 – h(6)
Empty
Empty

- При возникновении коллизии используется методика повторного хеширования
- Простейший случай – следующая свободная ячейка (линейная функция заполнения)
- Отрицательный эффект – длинные заполненные блоки («паровозы»)
- Как вариант: $h_i(x) = (h(x) + C \cdot i) \bmod B$
- Еще вариант: $h_i(x) = (h(x) + d_i) \bmod B$
- Удаление элемента – пометка его признаком deleted (не Empty)

Хеш-функция

- Хорошая функция
 - Быстрое вычисление
 - Минимальное количество коллизий
- Хеш-код – остаток от деления на B
- Хеш-код – сумма значений по модулю B

Основные требования к хеш-функциям

- Время вычисления
- Равномерность распределения значений при случайном выборе входного значения
- (Для криптографии) При равном изменении входных данных случайное изменение вычисленного хеш-кода
- Однонаправленность
- Устойчивость к коллизиям