

```

#include <iostream>
#include <vector>
#include <algorithm>
#include <cmath>

void radixSort(std::vector<int>& arr)
{
    // Определяем максимальное количество цифр в элементах массива
    int maxDigits = 0;
    for (int i = 0; i < arr.size(); i++) {
        int digits = 0;
        int n = arr[i];
        while (n > 0) {
            digits++;
            n /= 10;
        }
        if (digits > maxDigits) {
            maxDigits = digits;
        }
    }
    int powerlvl = 1; // степень разряда
    // Сортируем числа, начиная с младшего разряда
    for (int digit = 1; digit <= maxDigits; digit++) {
        // Создаем 10 пустых карманов
        std::vector<std::vector<int>> buckets(10);

        // Кладем элементы массива в соответствующие карманы на основе их
текущего значения разряда
        for (int i = 0; i < arr.size(); i++) {
            int bucketIndex = (arr[i] / powerlvl) % 10;
            buckets[bucketIndex].push_back(arr[i]);
        }
        powerlvl *= 10;
        // Объединяем все карманы в единый массив
        int index = 0;
        for (int i = 0; i < 10; i++) {
            for (int j = 0; j < buckets[i].size(); j++) {
                arr[index++] = buckets[i][j];
            }
        }
    }
}

int main()
{
    std::vector<int> arr {123, 45, 789, 25, 654, 987};

    std::cout << "Before sorting: ";
    for (int i = 0; i < arr.size(); i++) {
        std::cout << arr[i] << " ";
    }
    std::cout << '\n';

    radixSort(arr);
}

```

```

std::cout << "After sorting: ";
for (int i = 0; i < arr.size(); i++) {
    std::cout << arr[i] << " ";
}
//std::cin.ignore();
//std::cin.get()
return 0;
}
////////////////////////////////////
////////////////////////////////////

```

Сортировка LSD (наименее значащий разряд), карманная сортировка и сортировка подсчетом — все это алгоритмы сортировки с линейным временем, которые имеют некоторые сходства и различия:

LSD-сортировка — это алгоритм сортировки по основанию, который сортирует целые числа и строки, проверяя цифры или символы справа налево, начиная с младшей значащей цифры.

LSD-сортировка работает, сначала сортируя наименее значащую цифру, затем сортируя следующую наименее значащую цифру и так далее, пока не будет отсортирована самая старшая цифра.

Напротив, карманная сортировка и сортировка подсчетом не проверяют цифры или символы; они сортируют целые числа, распределяя их по корзинам или подсчитывая количество вхождений каждого целочисленного значения.

Карманная сортировка работает путем разделения входных данных на набор сегментов, а затем сортировки каждого сегмента по отдельности.

Сегменты обычно реализуются в виде массивов или связанных списков, а элементы распределяются по сегментам на основе некоторой функции значений элементов.

Затем сегменты сортируются по отдельности либо рекурсивно, либо с использованием другого алгоритма сортировки, а затем объединяются для формирования окончательного отсортированного вывода.

Карманная сортировка часто используется, когда входные данные равномерно распределены по диапазону значений, а количество элементов не слишком велико.

Сортировка подсчетом работает, подсчитывая количество вхождений каждого целочисленного значения во входных данных, а затем используя эту информацию для создания отсортированного вывода.

Сортировка подсчетом часто используется, когда входные данные состоят из относительно небольших и плотно упакованных целых чисел, таких как частота встречаемости символов в строке.

Таким образом, LSD-сортировка — это алгоритм сортировки по основанию, который сортирует целые числа и строки путем проверки цифр или символов, в то время как карманная и подсчетом — это алгоритмы сортировки на основе распределения,

которые сортируют целые числа, разделяя входные данные на сегменты или подсчитывая вхождения каждого значения.

Все три алгоритма имеют свои сильные и слабые стороны и могут использоваться в зависимости от конкретных характеристик входных данных.

Сортировка LSD (наименее значащий разряд) обычно используется, когда входные данные представлены в виде строк или целых чисел, а длина строк или диапазон целых чисел относительно малы.

Сортировка LSD имеет временную сложность $O(w * n)$, где w — максимальное количество цифр или символов во входных данных, а n — количество элементов во входных данных.

Напротив, карманная сортировка и сортировка подсчетом часто используются, когда входные данные состоят из целых чисел, которые равномерно распределены в известном диапазоне, а диапазон целых чисел относительно мал по сравнению с количеством элементов. Карманная сортировка и сортировка подсчетом имеют временную сложность $O(n + k)$, где n — количество элементов во входных данных, а k — диапазон целых чисел.

В общем, сортировка LSD хорошо подходит для сортировки строк или целых чисел с небольшим диапазоном значений и небольшим количеством цифр или символов.

Это связано с тем, что сортировка LSD сравнивает каждую цифру или символ входных данных, что делает ее эффективной для небольших значений. С другой стороны, карманная сортировка и сортировка подсчетом более эффективны,

когда диапазон целых чисел невелик, а количество элементов велико, поскольку они требуют меньше памяти и меньше сравнений, чем сортировка LSD.

Таким образом, сортировка LSD является хорошим выбором, когда входные данные представлены в виде строк или целых чисел с небольшим диапазоном значений и небольшим количеством цифр или символов, в то время как карманная сортировка и сортировка подсчетом больше подходят для равномерно распределенных целых чисел в небольшом диапазоне с большим количеством элементов.

Карманная сортировка по младшим разрядам a.k.a LSD_sort

created by Агафонов Олег 62 группа 2 курс

2023г.

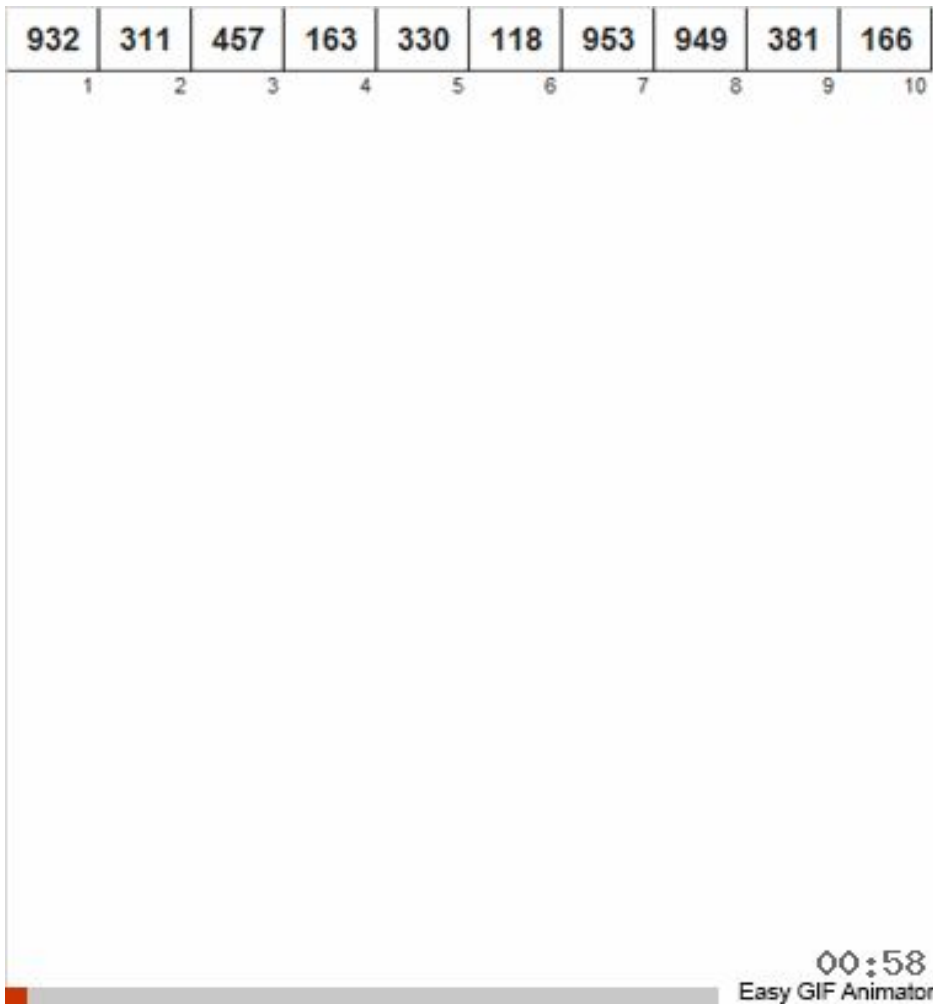
Алгоритмическая сложность

Time			Space
Worst case	Best case	Average case	Worst case
$O(wn)$	$O(wn)$	$O(wn)$	$O(n + r)$ auxiliary

Where:

- n = input size
- w = word size
- r = radix

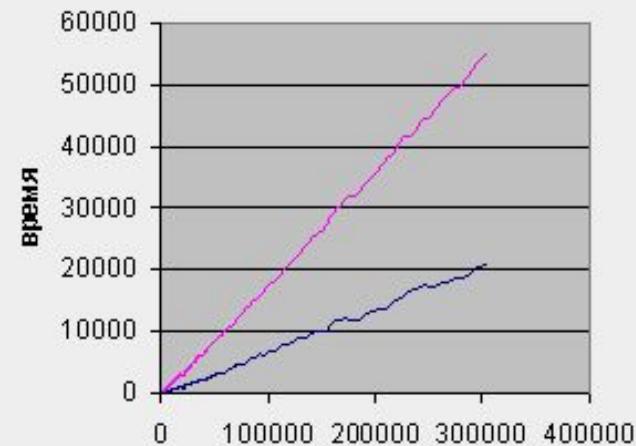




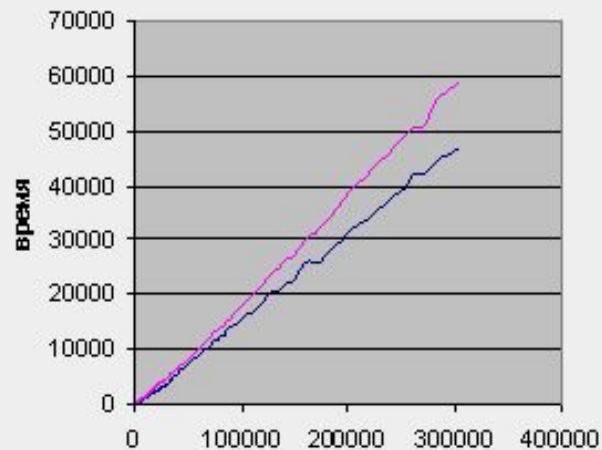
**Визуализация
через видео**

Тестирование

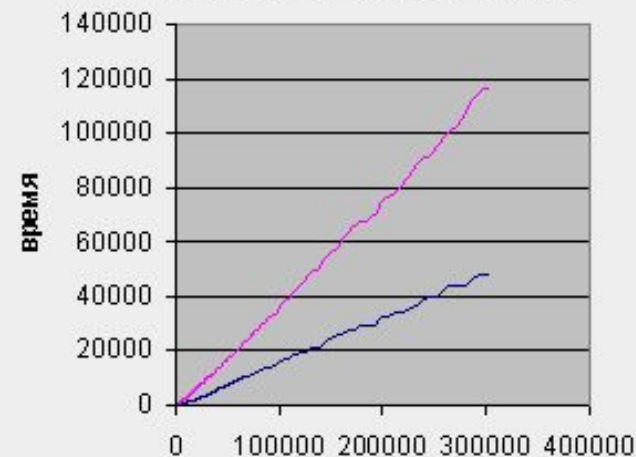
На диаграммах изображены результаты сравнений поразрядной сортировки (синяя линия) и быстрой (розовая линия), причем использовалась функция `sort()` из библиотеки STL.



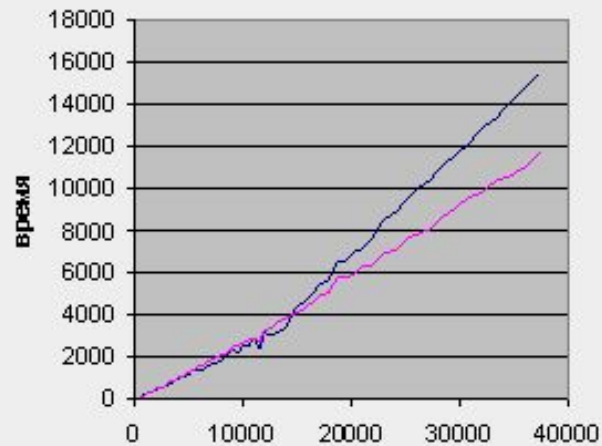
число элементов (2 байта: short)



число элементов (4 байта: long)



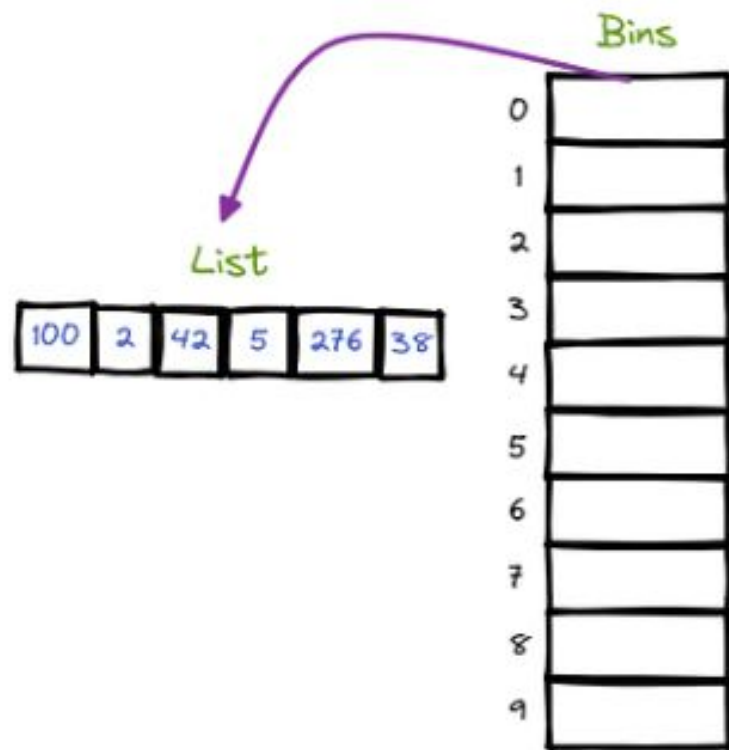
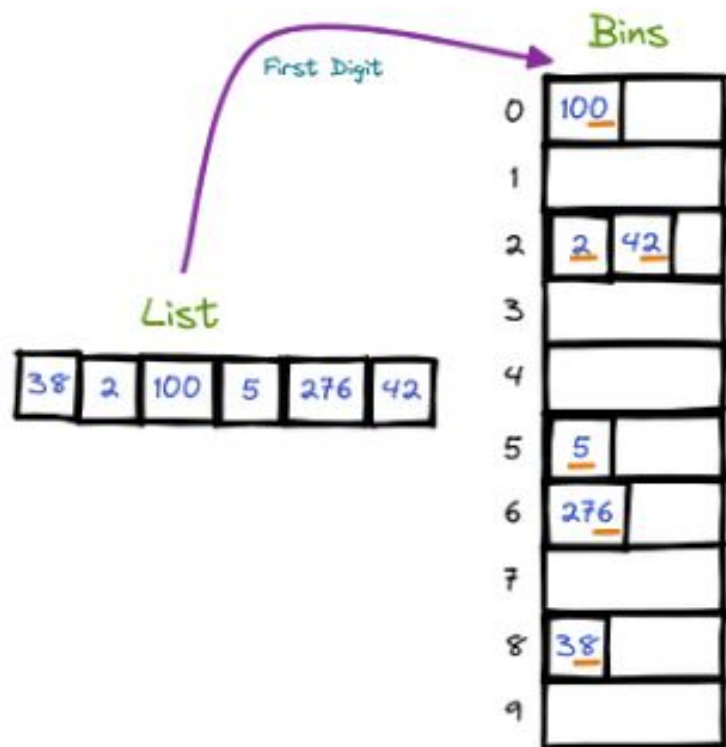
число элементов (4 байта: float)

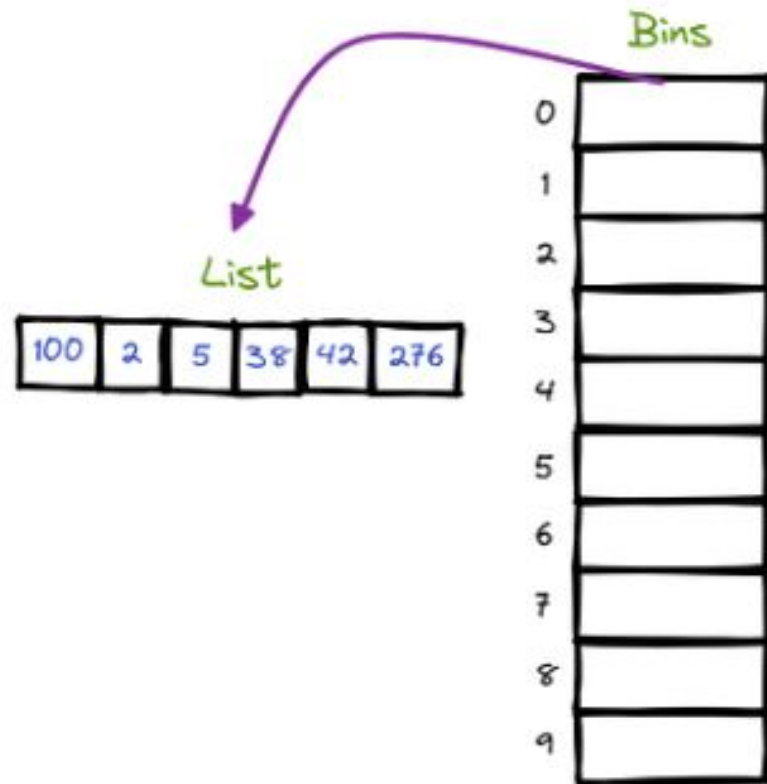
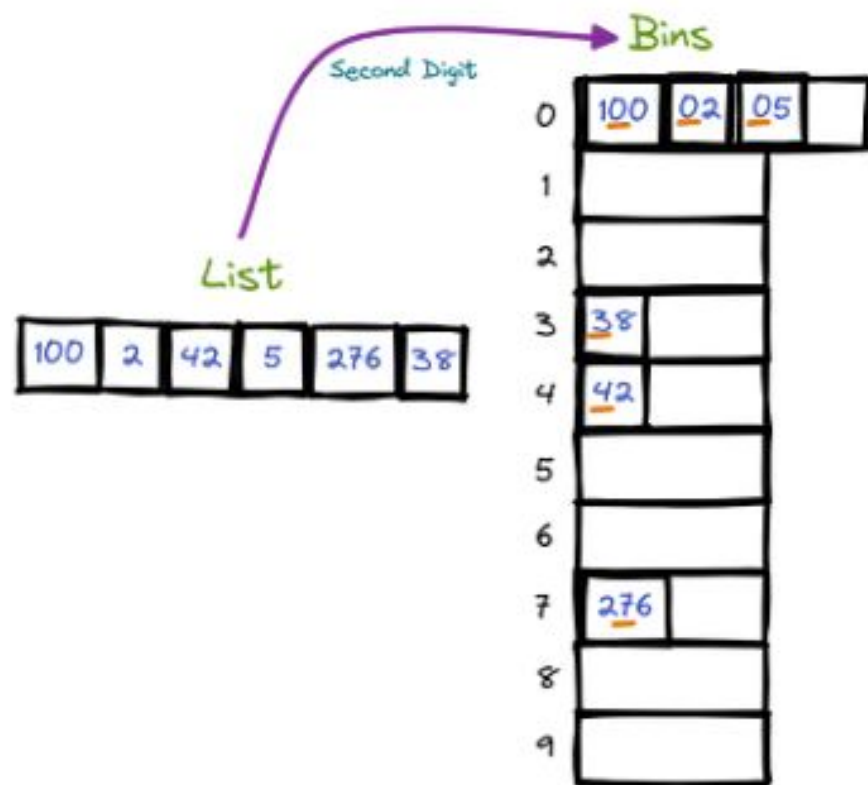


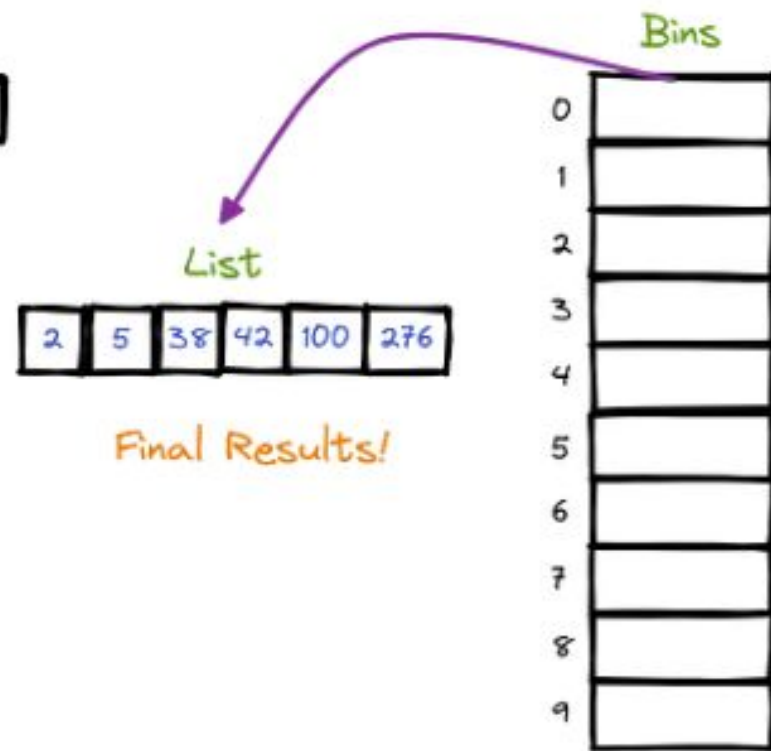
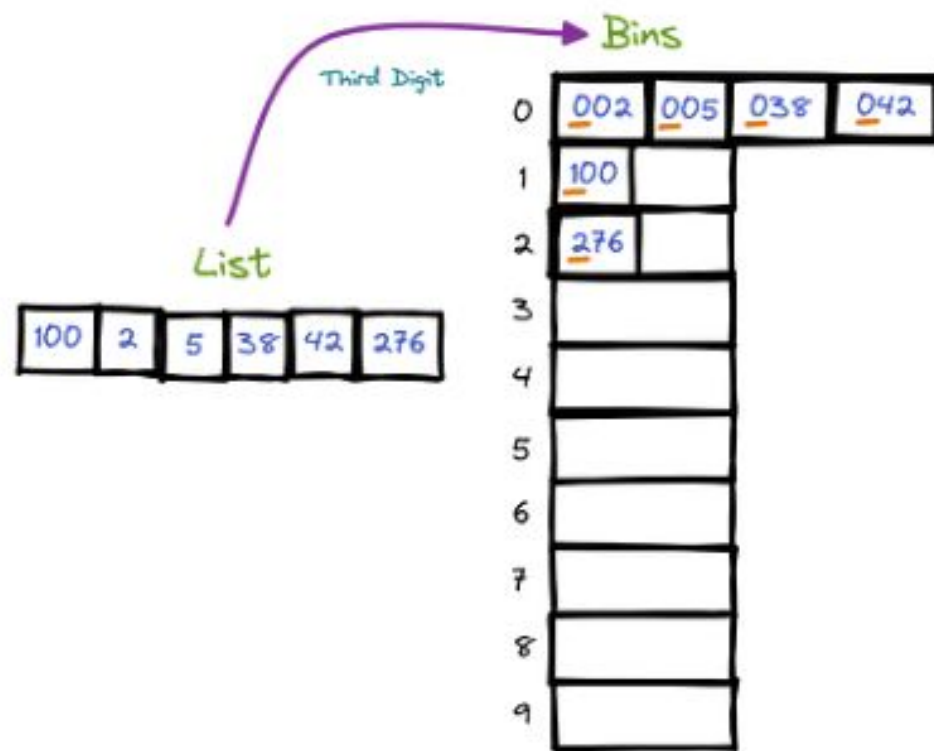
число элементов (8 байт: double)

Визуализация через картинки









THANK YOU

GRACIAS
ARIGATO
SHUKURIA
JUSPAXAR
DANKSCHEEN
TASHAKKUR ATU
YAQHANYELAY
SUKSAMA
EKHMET
BIYAN
SHUKRIA
TINGKI
GOZAIMASHITA
EFCHARISTO
PAIDIES
BOLZIN
MERCI
GRAZIE
MEHRBANI
MAAKE
KOMAPSUNNIDA
CHALTU
NUHUN
SHACHALHUYA
SPASSIBO
WADEEJA
MAITEKA
HUI
YUSPAGADATAM
UNALCHEESH
SPASIBO
DENKAUJA
HEHACHALHYA
ATTO
AMHA
DIHANYABAD
SAIKO
MERASTAHMY
GAEJTHO
AGUYJE
FAKAAUE
LAH
BAIKA
TAVTAPUCH
MEDAHAGSE
MINMONCHAR
MAKETAI
SIKOMO
EXOUJ
NATUR
GUI