

Университет ИТМО
Кафедра ВТ

Задачи 2 (2 - Шпион)

Алгоритмы и Структуры Данных

Порядок восстановления строки

Выполнил: Федоров Сергей
Группа: P3212

Санкт-Петербург
2020 г.

• Задача 2 - Шпион - 1322

Прелюдия: я знал что школьный курс по сжатию данных, мне когда-нибудь пригодится.

По сути шифрования исходной строки происходит путем преобразования Барроуза — Уилера. Тогда для того чтобы расшифровать закодированное сообщение, нужно применить обратное преобразование Барроуза — Уилера. Суть обратного преобразования заключается в последовательной сортировке исходного сообщения с последующим приписыванием слева его же самого:

Обратное преобразование			
Вход			
BNN . AA . A			
Добавление 1	Сортировка 1	Добавление 2	Сортировка 2
B	A	BA	AN
N	A	NA	AN
N	A	NA	A.
.	B	.B	BA
A	N	AN	NA
A	N	AN	NA
.B
A	.	A.	..

Добавление 3	Сортировка 3	Добавление 4	Сортировка 4
BAN	ANA	BANA	ANAN
NAN	ANA	NANA	ANA.
NA.	A..	NA..	A..B
.BA	BAN	.BAN	BANA
ANA	NAN	ANAN	NANA
ANA	NA.	ANA.	NA..
..B	.BA	..BA	.BAN
A..	..B	A..B	..BA

Добавление 5	Сортировка 5	Добавление 6	Сортировка 6
BANAN	ANANA	BANANA	ANANA.
NANA.	ANA..	NANA..	ANA..B
NA..B	A..BA	NA..BA	A..BAN
.BANA	BANAN	.BANAN	BANANA
ANANA	NANA.	ANANA.	NANA..
ANA..	NA..B	ANA..B	NA..BA
..BAN	.BANA	..BANA	.BANAN
A..BA	..BAN	A..BAN	..BANA

Добавление 7	Сортировка 7	Добавление 8	Сортировка 8
BANANA.	ANANA..	BANANA..	ANANA..B
NANA..B	ANA..BA	NANA..BA	ANA..BAN
NA..BAN	A..BANA	NA..BANA	A..BANAN
.BANANA	BANANA.	.BANANA.	BANANA..
ANANA..	NANA..B	ANANA..B	NANA..BA
ANA..BA	NA..BAN	ANA..BAN	NA..BANA
..BANAN	.BANANA	..BANANA	.BANANA.
A..BANA	..BANAN	A..BANAN	..BANANA

| Результат | | | |
| .BANANA. | | | |

Однако учитывая наш порядок N, это достаточно дорого каждый раз сортировать весь список, а так же хранить всю матрицу символов.

Существует оптимизированное обратное преобразование. Отсортируем сообщение один раз, запоминая их изначальные позиции. Затем используя известное число строки с правильным словом начнем с соответствующего символа, используя его индекс как ссылку к следующему символу, и так повторим N раз.

Вопрос: В каком порядке восстанавливается строчка?

Пусть данная строка: B N N <SOS> A A <EOS> A
 1 2 3 4 5 6 7 8

После сортировки получаем: A A A B N N <SOS> <EOS>
 5 6 8 1 2 3 4 7

Зная <SOS>-символ начинаем последовательно выводить символы, выбирая следующий символ по индексу текущего.

<SOS> (4) → B (1) → A (5) → N (2) → A (6) → N (3) → A (8) → <EOS>

Результат: <SOS> B A N A N A <EOS>

Ответ: Порядок восстановления строчки - **прямой**.

Почему работает?

Вспомним изначальный способ обратного преобразования. На последнем шаге преобразования мы имели бы такую матрицу символов:

Наша строка после сортировки

|

v

v- Изначально данная строка

```

ANANA . . B
ANA . . BAN
A . . BANAN
BANANA . .
NANA . . BA
NA . . BANA
. BANANA .
. . BANANA
  
```

```

1.  N>|
    | | N>|
2   | | | |
    | | | |
3   | | | | N>|
    | | | | |
4  N>| | | | |
    | | | | |
5   | N>| | | |
    |   | | |
6   |   | | |
    |   | |
7  >|   |
    |   |
8.           N
  
```

Каждый раз когда мы переходим к следующему символу, по сути переходим к номеру строки следующей циклической перестановки в нашей матрице (матрица состоит из всех циклических перестановок). Берем первый символ (так как нам известен только первый и последний столбец) и идем к следующей строке.

Почему при взятии строки по индексу получается следующая циклическая перестановка. Потому что имея по сути два столбца (первый и последний), при взятии по индексу некоторого символа мы идем к строке где этот элемент будучи первым стал последним (циклическая перестановка влево).

Код:

```

//
// Created by Sergey Fedorov on 14/03/2020.
//
#include <iostream>
#include <cmath>
#include <algorithm>

using namespace::std;

bool letter_comprar(pair<char, int> a, pair<char, int> b){
    if (a.first == b.first){
        return a.second < b.second;
    } else {
        return a.first < b.first;
    }
}
  
```

```

int main() {

    int k;
    cin >> k;

    string encoded;
    cin >> encoded;

    int n = encoded.length();
    pair<char, int> letter_2_position[n];

    for (int i = 0; i < n; ++i) {
        letter_2_position[i] = (pair<char, int>) {encoded[i], i};
    }

    sort(letter_2_position, letter_2_position + n, letter_comprar);

    int back_i = k - 1;
    for (int i = 0; i < n; ++i) {
        pair<char, int> next_letter = letter_2_position[back_i];
        cout << next_letter.first;
        back_i = next_letter.second;
    }
    cout << endl;
}

```