

УДК 681.3.06

# ПРОГРАММА ВНУТРЕННЕЙ СОРТИРОВКИ ДЛЯ ЭВМ "МИНСК-32"

Н.С.Кормилицын, Ю.Г.Косарев

НАИМЕНОВАНИЕ - Сортировка массива внутренняя (ВНУС).

РЕДАКЦИЯ - 1972 года.

ЯЗЫК ПРОГРАММИРОВАНИЯ - ЯСК.

НАЗНАЧЕНИЕ - программа предназначена для упорядочения размещенного в ОЗУ массива однотипных многоразрядных фраз по признакам, произвольно распределенным по фразе.

ИСХОДНЫЕ ДАННЫЕ:

- а) параметры массива и указатель режима сортировки (по возрастанию или убыванию значений признаков);
- б) начальный адрес массива фраз, размещенного в ОЗУ;
- в) начальный адрес таблицы выделителей признаков.

РЕЗУЛЬТАТ - массив фраз, упорядоченный по заданным признакам и размещенный в том же месте ОЗУ.

ОБРАЩЕНИЕ НА ЯСК

| Этикетка | КОП | Адреса и замечания |
|----------|-----|--------------------|
|          | ИП  | СОРТ;6             |
|          | КЧ  | М                  |
|          | КЧ  | ↓                  |
|          | КЧ  | Z                  |
|          | КЧ  | K                  |
|          | КЧ  | P                  |
|          | КА  | MAC;TV             |

Здесь

| Этикетка | КОП        | Адреса и замечания |
|----------|------------|--------------------|
| СОРТ     | ОПР<br>НОП | ВНУС               |

M, Q, Z, K - восьмеричные числа;

M - количество фраз в массиве, размещенном в ОЗУ;

Q - количество ячеек во фразе;

Z - количество признаков, по которым выполняется сортировка;

K - количество двоичных разрядов в признаковом слове, формируемом из признаков;

P - указатель режима сортировки по возрастанию ( $P=1$ ) или убыванию ( $P=0$ ) значений признаков;

MAC - начальный адрес массива фраз;

TV - начальный адрес таблицы выделителей признаков, элементы которой имеют следующий вид:

| Этикетка | КОП | Адреса и замечания |
|----------|-----|--------------------|
|          | КЧ  | $\alpha_1$         |
|          | КЧ  | $V_j$              |

$\alpha_1$  - восьмеричный порядковый номер ячейки во фразах, где находится признак, выделяемый с помощью выделителя  $V_j$ ;

$V_j$  - константа числовая восьмеричная (выделитель) для выделения признака с номером j.

Количество элементов таблицы TV равно количеству признаков. Таблица строится по возрастанию старшинства признаков.

ПАМЯТЬ. Программа занимает

- 217 ячеек основного поля,

- 16 ячеек индексного поля,

- динамически резервирует рабочий участок из M ячеек для массива признаков слов и рабочий участок из  $2^K + 1$  ячеек для массива счетчиков.

АЛГОРИТМ

Идея алгоритма основана на построении перестановки  $P=(x_1, \dots, x_n, \dots, x_n)$ , которая позволяет путем однократного перемещения каждой фразы с места  $x_n$  на место n полностью упорядочить массив. Перестановка P получается как произведение перестановок  $P_r$ , соответствующих ключевым (признаковым) словам  $k_1, \dots, k_r$ . Каждое ключевое слово составляется из подряд расположенных ключей (частичных признаков) и имеет заданную длину k.

Перестановки  $P_r$  получаются следующим путем:

- образуется массив  $b[i]$  значений признаков, соответствующих ключевому слову  $K_r$ ;
- для всех  $b[i]$  определяется распределение  $c[t]$  по частоте встречаемости, где  $t = 0, 1, \dots, 2^k - 1$ ;
- для каждого  $t$  определяются границы  $y[t]$  для номеров мест в массиве, упорядоченном по  $K_r$  ключевому слову

$y[0] := 0$

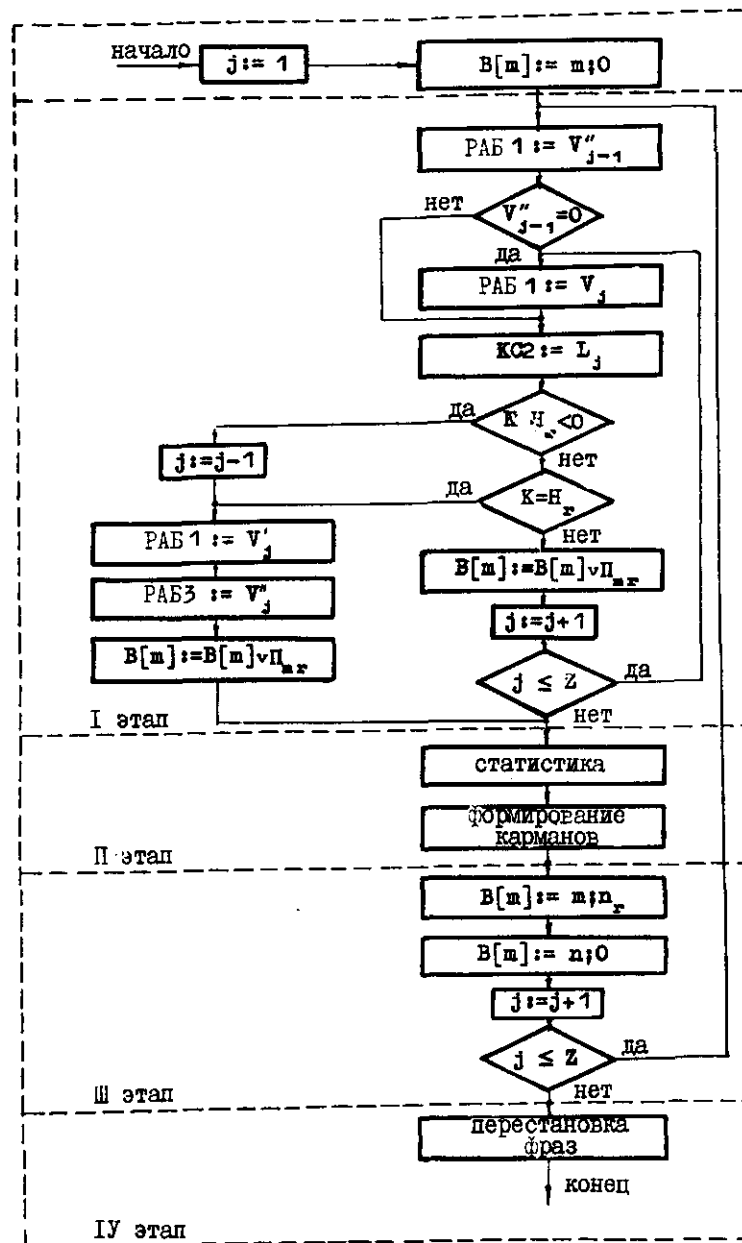
$y[t] := y[t-1] + c[t-1]$ ;

- для каждого  $b[i]$  находится соответствующий элемент перестановки  $P_r$ ,  $x_i = y[b[i]] + r - 1$ , где  $r - 1$  - число предшествующих обращений к  $y[t]$ .

Программа состоит из четырех этапов (см. рисунок). При входе в программу в массиве признаков слов формируется исходная информация - по первому адресу каждой из  $M$  ячеек записывается текущее значение адреса (от 0 до  $M-1$ ). При этом в массиве признаков слов каждой фразе сортируемого массива будет соответствовать одна ячейка. Младшие разряды второго адреса в количестве  $K$  выделяются для формирования собственно массива признаков слов.

**Этап I.** По таблице выделителей признаков программа последовательно выделяет признаки в фразах и составляет из них признаковые слова  $\Pi_{rx}$  длиной  $K$  разрядов каждое. Заполнение признакового слова начинается с младших разрядов. При этом для каждого признака определяется своя константа сдвига  $L_j$ . В том случае, когда в признаковое слово не укладывается целое число признаков ( $N_r > K$ , где  $N_r$  - суммарная длина признаков, формирующих текущее признаковое слово), переполняющий признак разделяется на две части  $V'_j$  и  $V''_j$ . Первая часть признака  $V'_j$  входит в формируемое текущее признаковое слово, а оставшаяся часть  $V''_j$  образует начало очередного признакового слова. Поэтому при каждом входе в I этап производится проверка на остаток от предыдущего выделителя ( $V_{j-1} = 0$ ).

**Этап II.** По значениям признаков слов (используемых как адреса) в массиве счетчиков подсчитывается количество слов с одинаковыми признаками (статистика), и формируются "карманы" для всех подсчитанных значений признаков, то есть определяется количество и место в массиве для элементов, имеющих одинаковые значения признаков.



Блок-схема программы упорядочения массива

Э т а п III. По результатам этапа II строится очередная подстановка  $P_2$ , на которую умножается подстановка, полученная по предыдущему признаковому слову.

Далее происходит возврат к этапу I, формирование нового признакового слова из очередных, более старых признаков и новой таблицы перестановок. Этот процесс продолжается до полного исчерпания всех признаков. Последняя полученная таблица перестановок используется для перестановки фраз в массиве.

Э т а п IV. По таблице перестановок выполняется перестановка фраз массива. При этом после формирования текущего адреса проверяется наличие метки (фраза уже перестановлена ранее). При наличии метки увеличивается на единицу значение текущего адреса. При отсутствии метки выполняется проверка на совпадение адреса с адресом перестановки. Если адреса совпадают, значение счетчика перестановленных фраз увеличивается на единицу и формируется очередное значение текущего адреса. При несовпадении производится перестановка фраз. Для этого фраза с текущим адресом переставляется в рабочее поле (на место бывшего массива счетчиков), а на ее место в массиве перемещается фраза, имеющая соответствующий адрес перестановки. Перестановленная на место в массиве фраза метится (в массиве признаков слов), значение счетчика фраз увеличивается на единицу, далее проверяется совпадение адреса освободившегося места с текущим адресом. При несовпадении на освободившееся место в массиве перемещается фраза с соответствующим адресом перестановки. Такая перестановка фраз в массиве продолжается до совпадения очередного освободившегося места с адресом (текущим) фразы, находящейся в рабочем поле. В последнем случае фраза из рабочего поля пересылается на освободившееся место в массиве. Таким образом, завершается перестановка очередной последовательности (кольца) фраз. Далее текущий адрес снова увеличивается на единицу, и описанный процесс продолжается до завершения перестановки всех фраз массива.

Время работы программы приближенно можно определить по формуле:

$$T \leq \left[ \left( 25 \frac{N}{K} + 6,5Z + \frac{Q}{2} + 12 \right) M + 4,7 \cdot 2^{\frac{K}{K}} \frac{N}{K} \right] 30 \cdot 10^{-6} \text{ сек.}$$

где  $N$  — суммарная длина признаков.

Как видно из формулы, в общем времени работы программы значительный вес имеет время, определяемое вторым слагаемым в квадратных скобках и характеризующее время предварительной обработки признаков до упорядочения. Отсюда следует, что область предпочтительного применения программы является сортировка массивов, имеющих достаточно большой размер фраз ( $Q \leq 5$ ).

#### СООБЩЕНИЯ И УКАЗАНИЯ ОПЕРАТОРУ

| Текст сообщения     | Содержание                                                                                             | Решение оператора                                  | Директивы ответа оператора |
|---------------------|--------------------------------------------------------------------------------------------------------|----------------------------------------------------|----------------------------|
| НЕТ ПАМЯТИ ДЛЯ ВНУС | При динамическом резервировании памяти для вспомогательных массивов не хватает заданного объема памяти | Увеличить заказ на объем памяти, повторить задание | *X 0                       |