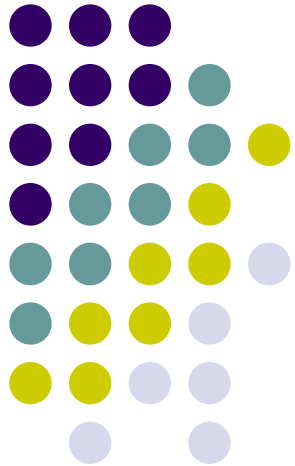
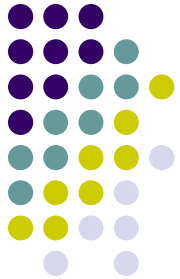
Символьные массивы. Поиск подстроки в строке. БМ-поиск, КМП - поиск





Дано. Текст s длины **n** и некоторая строка **p** длины m - *образец*.

Найти вхождение (первое) строки **p** в текст s. Определить позицию вхождения.

Рассмотрим 3 метода поиска.

1. Прямой поиск

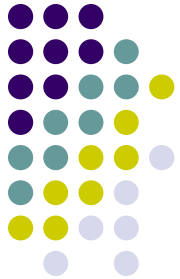
s = “во дворе трава на траве дрова”

p = “дрова”

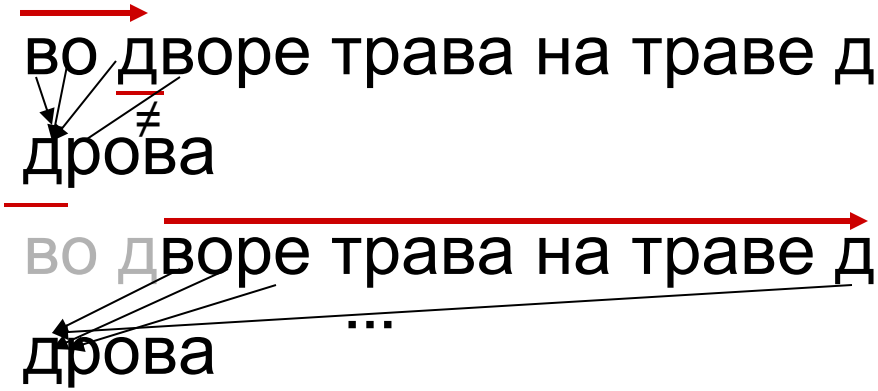
При прямом поиске последовательно идем по тексту **s**, пока не встретится символ, совпадающий с первым символом образца **p**.

Сравниваем последующие символы по длине образца.

Если какой-то символ текста и образца не совпал, сдвигаемся в тексте **s** *на одну позицию* от начала сравнения с первым символом образца, и все повторяем.

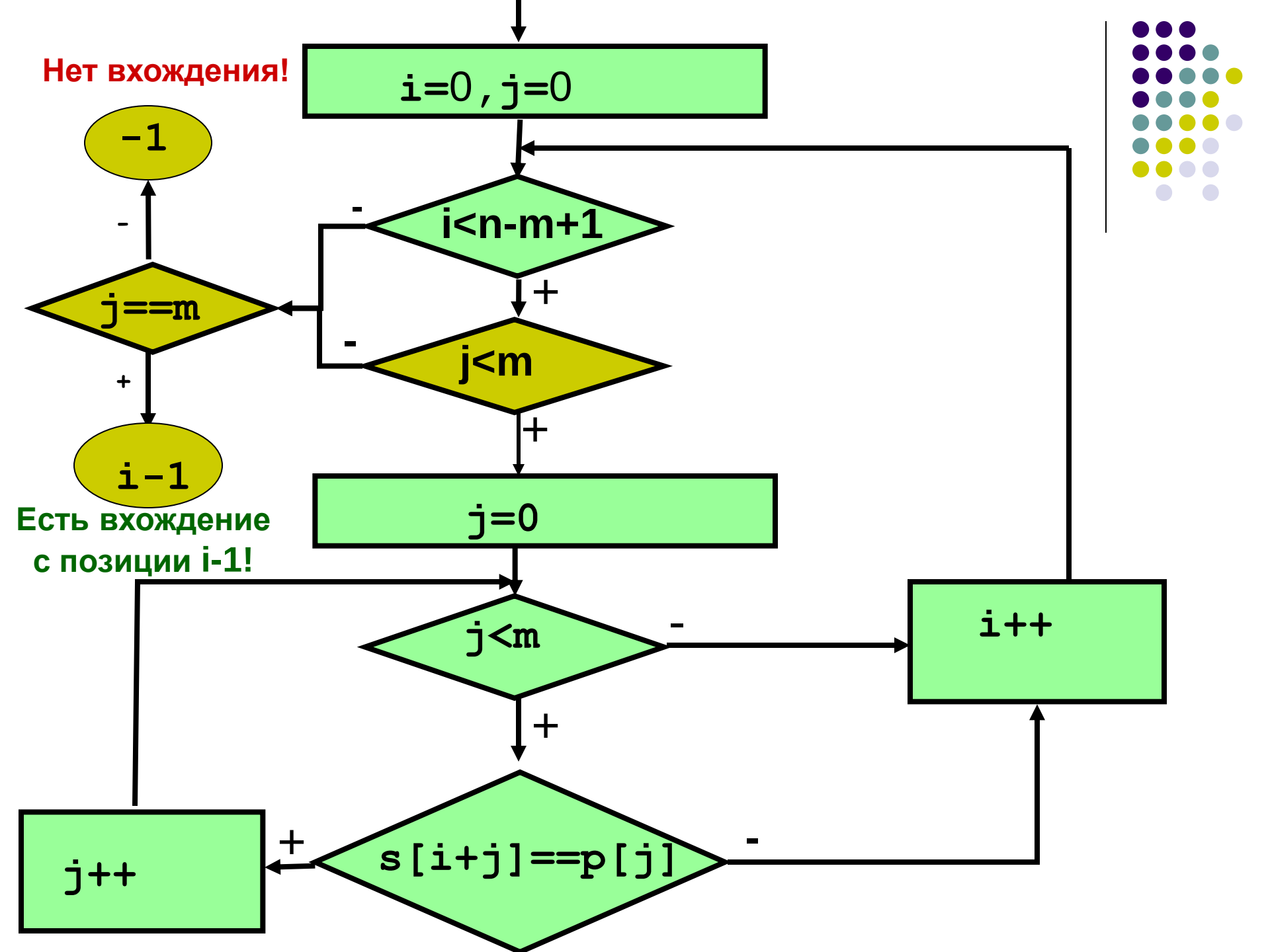


Пример



Для сравнения с другими методами важно заметить:

при неудачном сравнении – поиск продолжаем всегда с начала образца.


Нет вхождения!
i=0,j=0
-1
-
i<n-m+1
-
+
j==m
-
j<m
+
+
i-1
Есть вхождение
с позиции i-1!
j=0
j<m
-
i++
+
j++
+
s[i+j]==p[j]
-

Обсуждение

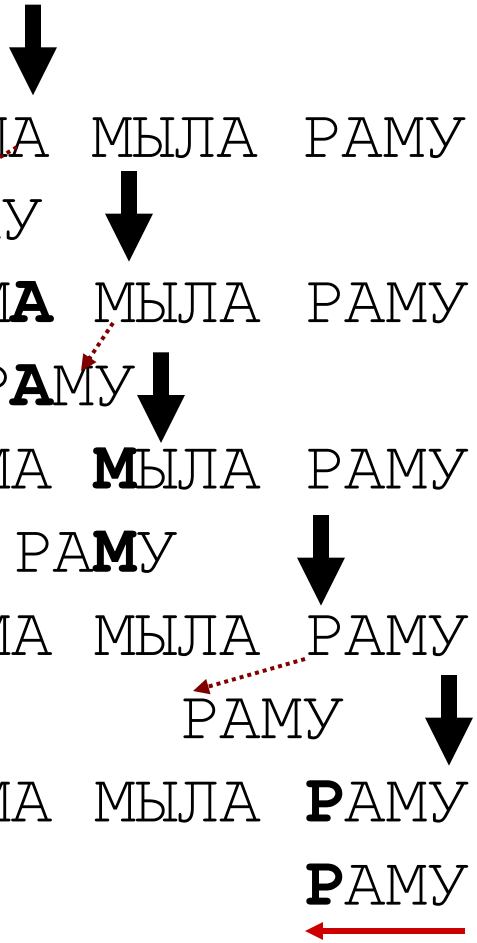
В прямом методе поиска **не** анализируется структура образца.

В последующих 2-х методах предварительный анализ образца (предтрансляция) позволяет сформировать некоторую полезную информацию о его структуре.

В процессе поиска эта информация используется для того, чтобы некоторые операции сравнения ***исключить***.

2. БМ-поиск (Бауэр и Мур)

S="МАМА МЫЛА РАМУ"
P="РАМУ"



А -- 2

М -- 1

Ы -- 4

Р -- 3

Сравнение с конца образца

А≠У, но в образце есть А, совместим их

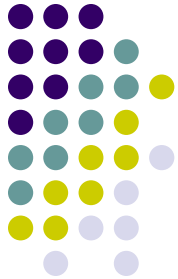
М≠У, но в образце есть М, совместим их

Ы≠У, и в образце нет Ы, сместимся на всю длину образца

Р≠У, но в образце есть Р, совместим их

Нашли!

Идея алгоритма



Числа 2, 1, 3 – это расстояния символов образца до его конца.

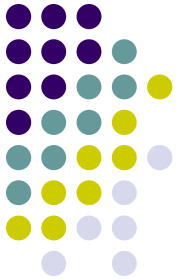
Таким образом, заранее готовится **таблица чисел из 256 элементов**, в которой формируются расстояния символов образца (кроме последнего) до конца образца.

Если каких-то символов **нет** в образце, то соответствующий элемент таблицы полагается **равным длине образца**(в нашем примере 4 – на столько происходит смещение в этом случае).

Для этого **все элементы** таблицы сначала полагаются равными длине образца **m**.

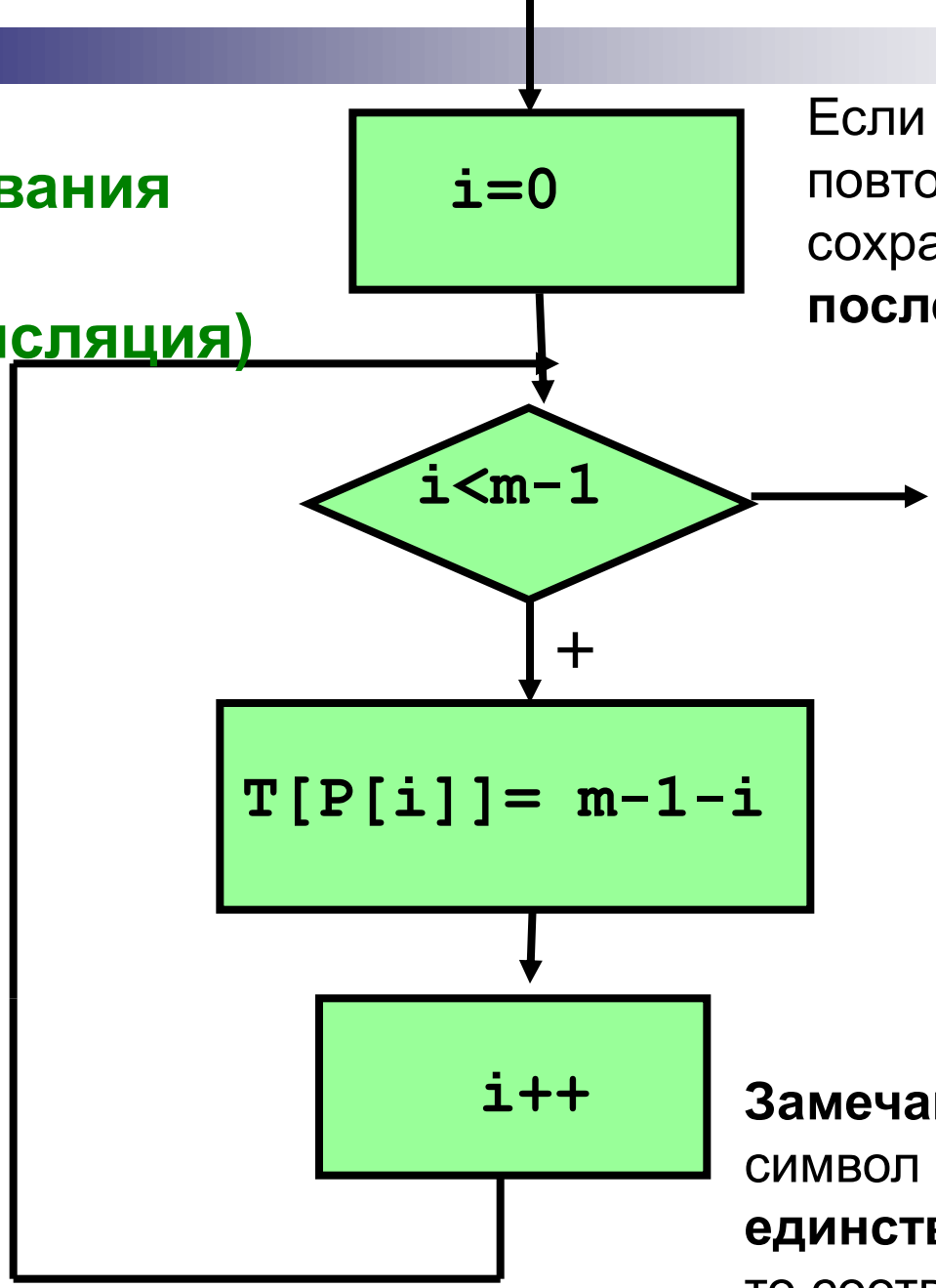
При поиске смещение добавляется к позиции той буквы текста **s**, которая соответствует последней букве образца на текущем шаге.

Таблица



Символу образца соответствует элемент таблицы с номером, равным коду символа!

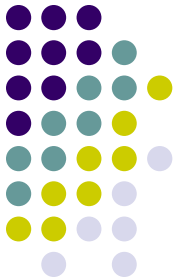
Алгоритм формирования таблицы (предтрансляция)



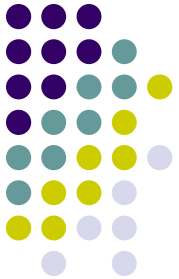
Если символ в образце повторяется, то в таблице сохраняется расстояние **последнего** из них.

Замечания. Если последний символ образца в нем **единственный**,
то соответствующий элемент таблицы остается равным **m**.

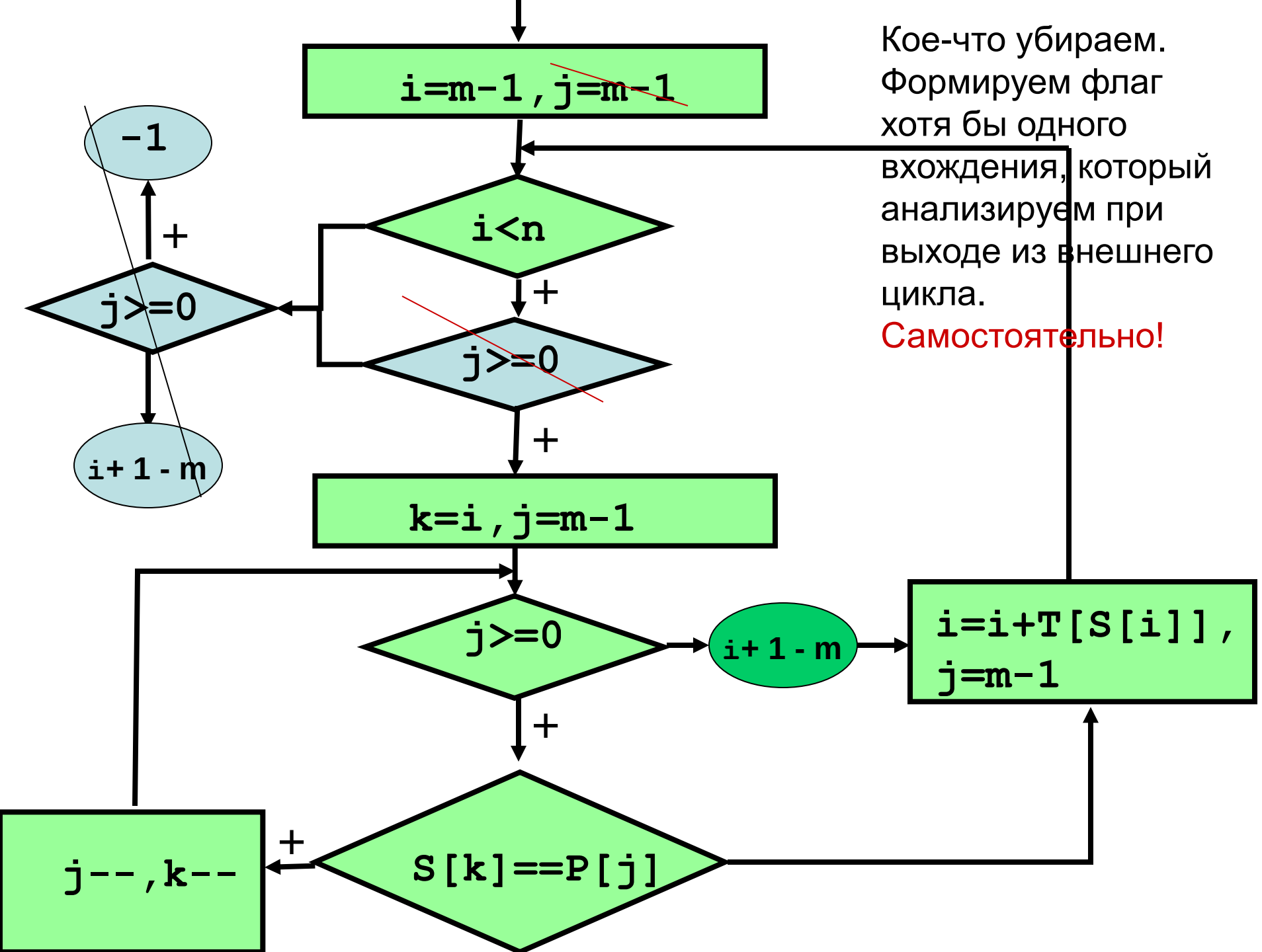
БМ-поиск



Замечание.



Для нахождения всех вхождений образца алгоритм модифицируется немного: после нахождения образца выводится позиция его начала (в нашей блок-схеме после внутреннего цикла) и поиск продолжается аналогично.



Кое-что убираем.
Формируем флаг хотя бы одного вхождения, который анализируем при выходе из внешнего цикла.
Самостоятельно!

Пример из 1.

s = “во дворе трава на траве дрова”

p = “дрова”

Таблица

_	…	а	б	в	г	д	…	о	р
5	…	5	5	1	5	4	…	2	3

+1

s = во дворе трава на траве дрова

p = дрова

+2

s = во дворе трава на траве дрова

p = дрова

e нет в образце!

+5

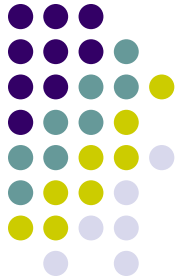
s = во дворе трава на траве дрова

p = дрова

+1

s = во дворе трава на траве дрова

p = дрова



+5

s = во дворе трава на траве

р = дрова

+5

s = во дворе трава на траве дрова

т нет в образце!

р = дрова

+5

s = во дворе трава на траве_дрова

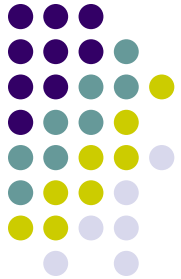
пробела нет в образце!

р = дрова

s = во дворе трава на траве_дрова

р = дрова ☺

_	…	а	б	в	г	д	…	о	р
5	…	5	5	1	5	4	…	2	3



8 новых позиций индекса i вместо 25 в прямом поиске!