

Тема 8. Алгоритмы и структуры данных. Поиск.

Алгоритмы поиска



Код	Наименование	Цена
44	Яблоки	35.50
55	Апельсины	29.90
12	Бананы	22.00
...

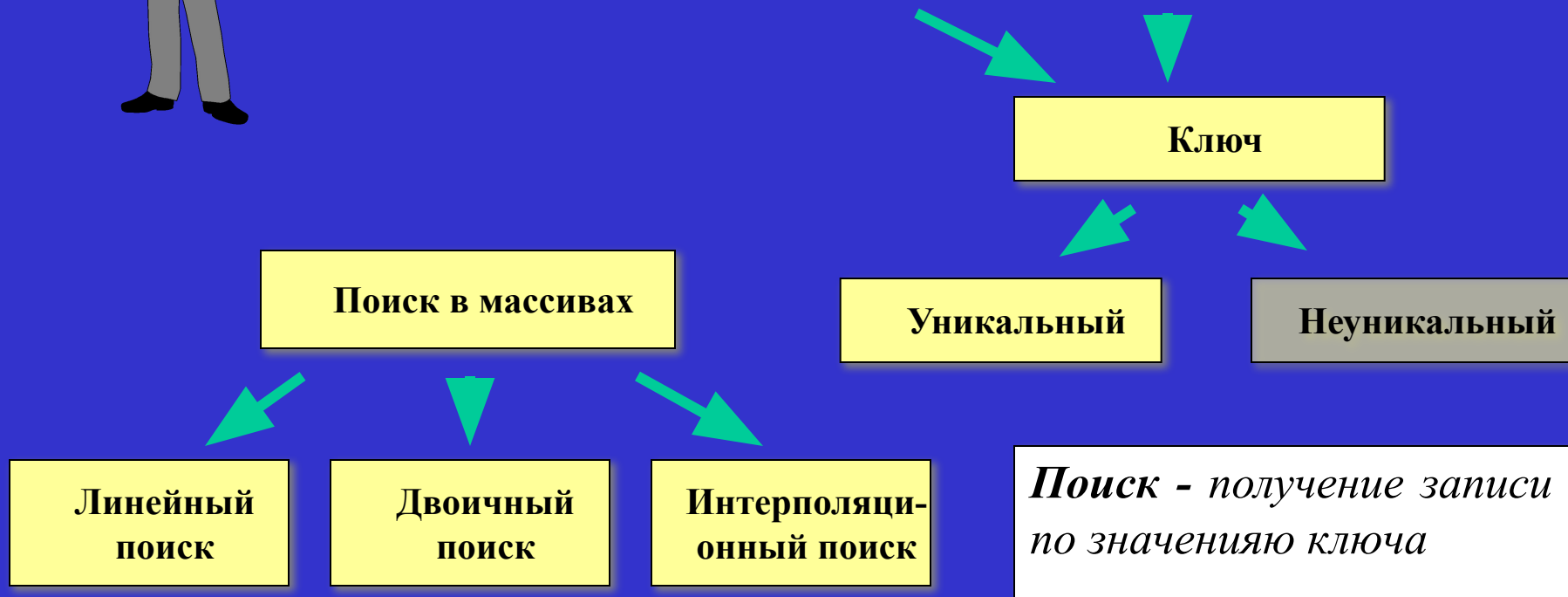
А. С. Пушкин
"ЕВГЕНИЙ ОНЕГИН"
ГЛАВА ПЕРВАЯ
«Мой дядя самых честных правил,
Когда не в шутку занемог,
Он уважать себя заставил
И лучше выдумать не мог.
Его пример другим наука;
Но, боже мой, какая скука
С больным сидеть и день и ночь,
Не отходя ни шагу прочь!
Какое низкое коварство
Полуживого забавлять,
Ему подушки поправлять,
Печально подносить лекарство,
Вздыхать и думать про себя:
Когда же черт возьмет тебя!»



Поиск в массивах

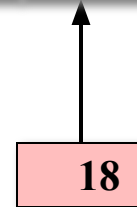


Код	Наименование	Цена
44	Яблоки	35.50
55	Апельсины	29.90
12	Бананы	22.00
...



Линейный поиск в массиве

01	99	44	55	12	42	94	18	06	67	98	03	95
----	----	----	----	----	----	----	----	----	----	----	----	----



*Линейный поиск - единственно
возможный способ поиска в
неупорядоченном массиве*

Эффективность $n/2$

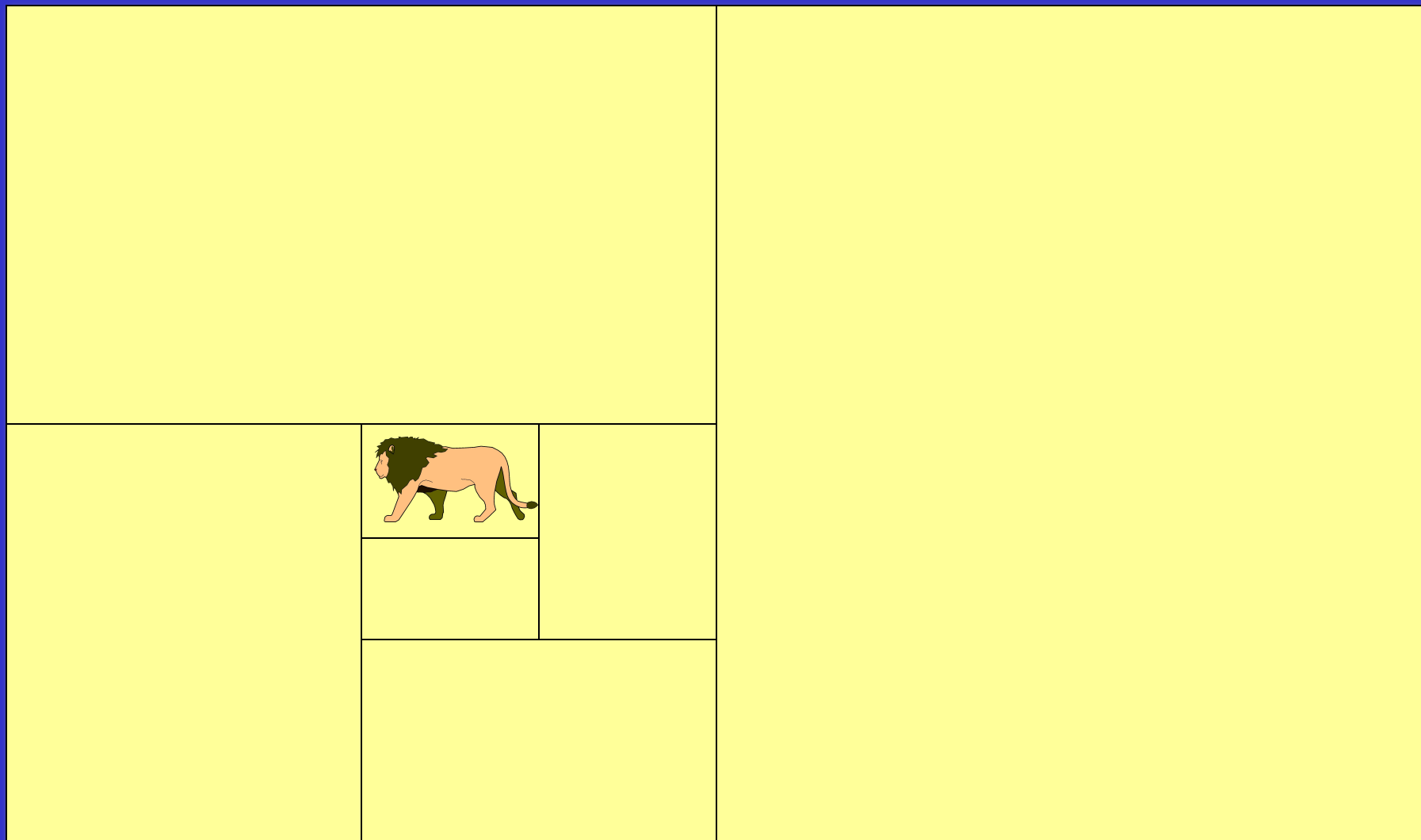
Пример линейного поиска в массиве

```
int LinearSearch(PROD* p, int n, short Val)
{
    for(int i = 0; i < n; i++)
        if(p[i].code == Val)
            return(i);

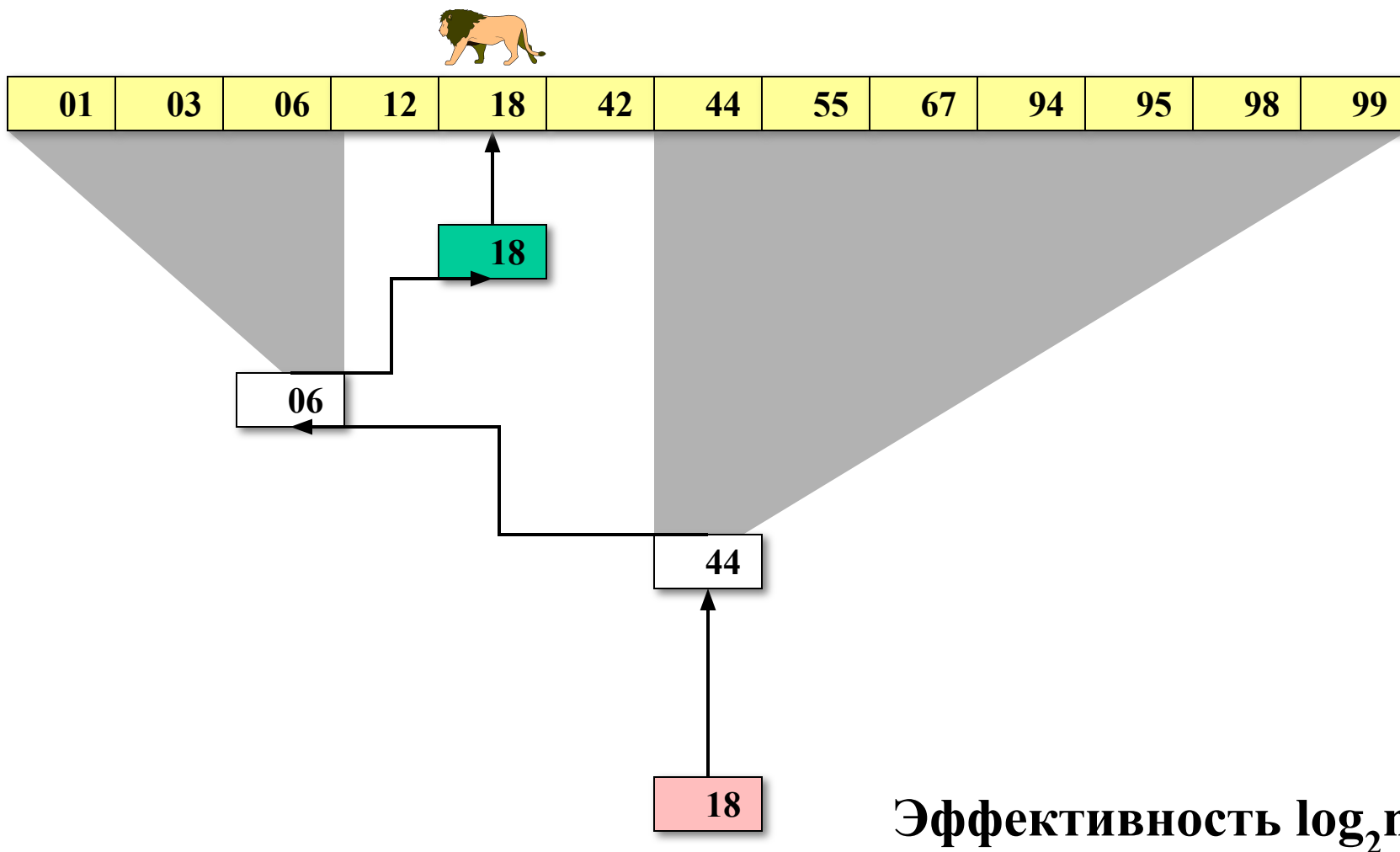
    return(-1);
}
```



Двоичный поиск в массиве



Двоичный поиск в массиве



Пример двоичного поиска в массиве

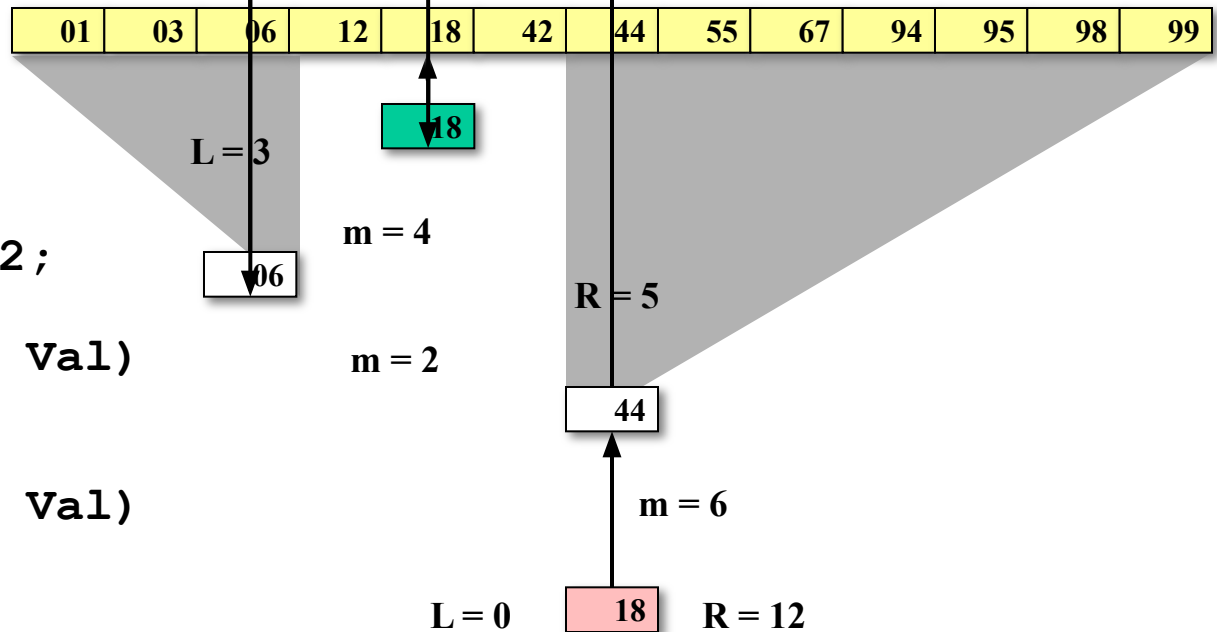
```

int BinarySearch(PROD* p, int n, short Val)
{
    int L = 0;
    int R = n-1;

    while(L <= R)
    {
        int m = (L+R)/2;

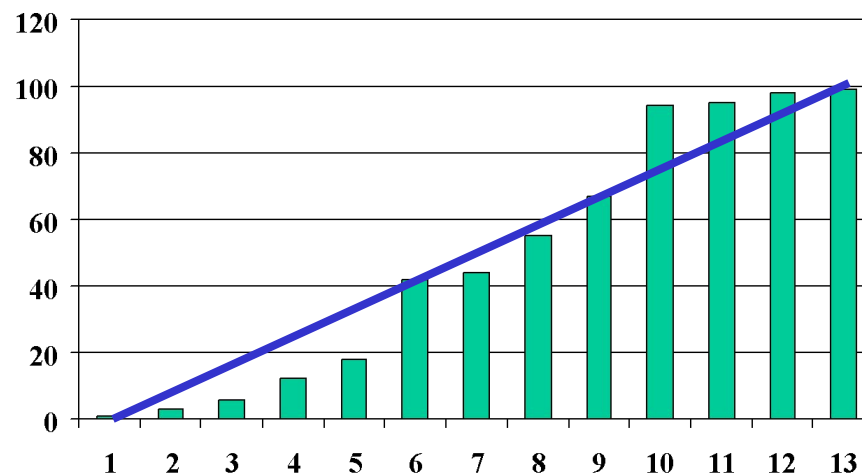
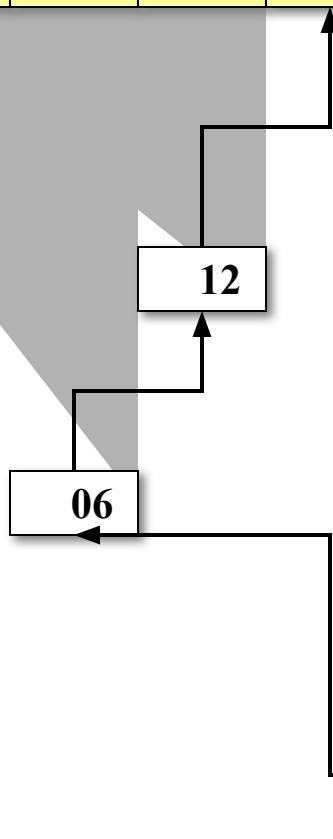
        if(p[m].code < Val)
            L = m+1;
        else
            if(p[m].code > Val)
                R = m-1;
            else
                return (m) ;
    }

    return (-1) ;
}
    
```



Интерполяционный поиск в массиве

01	03	06	12	18	42	44	55	67	94	95	98	99
----	----	----	----	----	----	----	----	----	----	----	----	----



$$\text{int ind} = L + ((V - a[L]) * (R - L)) / (a[R] - a[L]);$$

Эффективность $\log_2 n$

Пример интерполяционного поиска в массиве

```

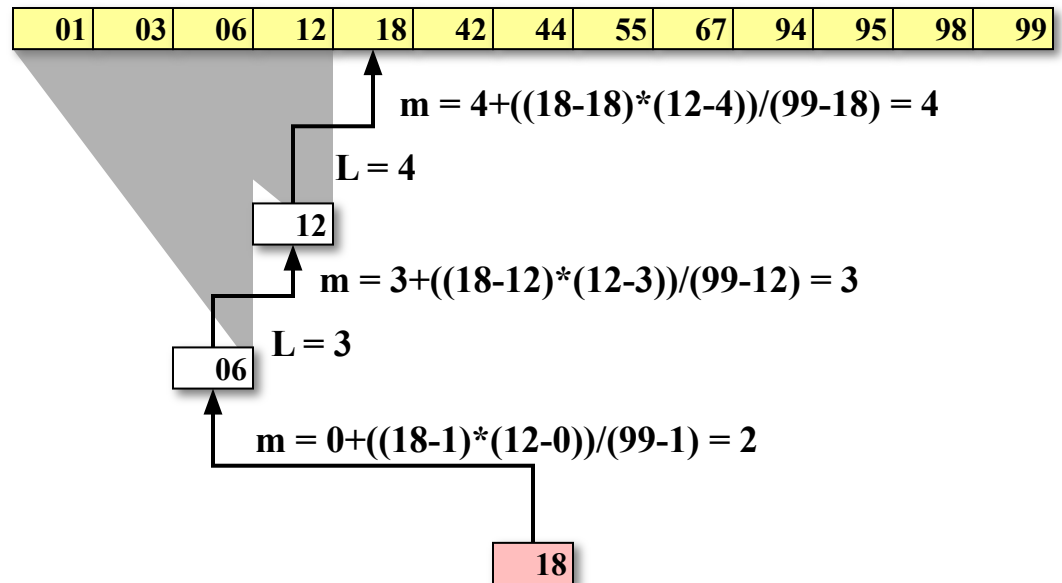
int InterpolationSearch(PROD* p, int n, short Val)
{
    int L = 0;
    int R = n-1;

    while(L <= R)
    {
        int m = L + ((Val - p[L].code) * (R - L)) / (p[R].code - p[L].code);

        if(p[m].code < Val)
            L = m + 1;
        else
            if(p[m].code > Val)
                R = m - 1;
            else
                return(m);
    }

    return(-1);
}

```



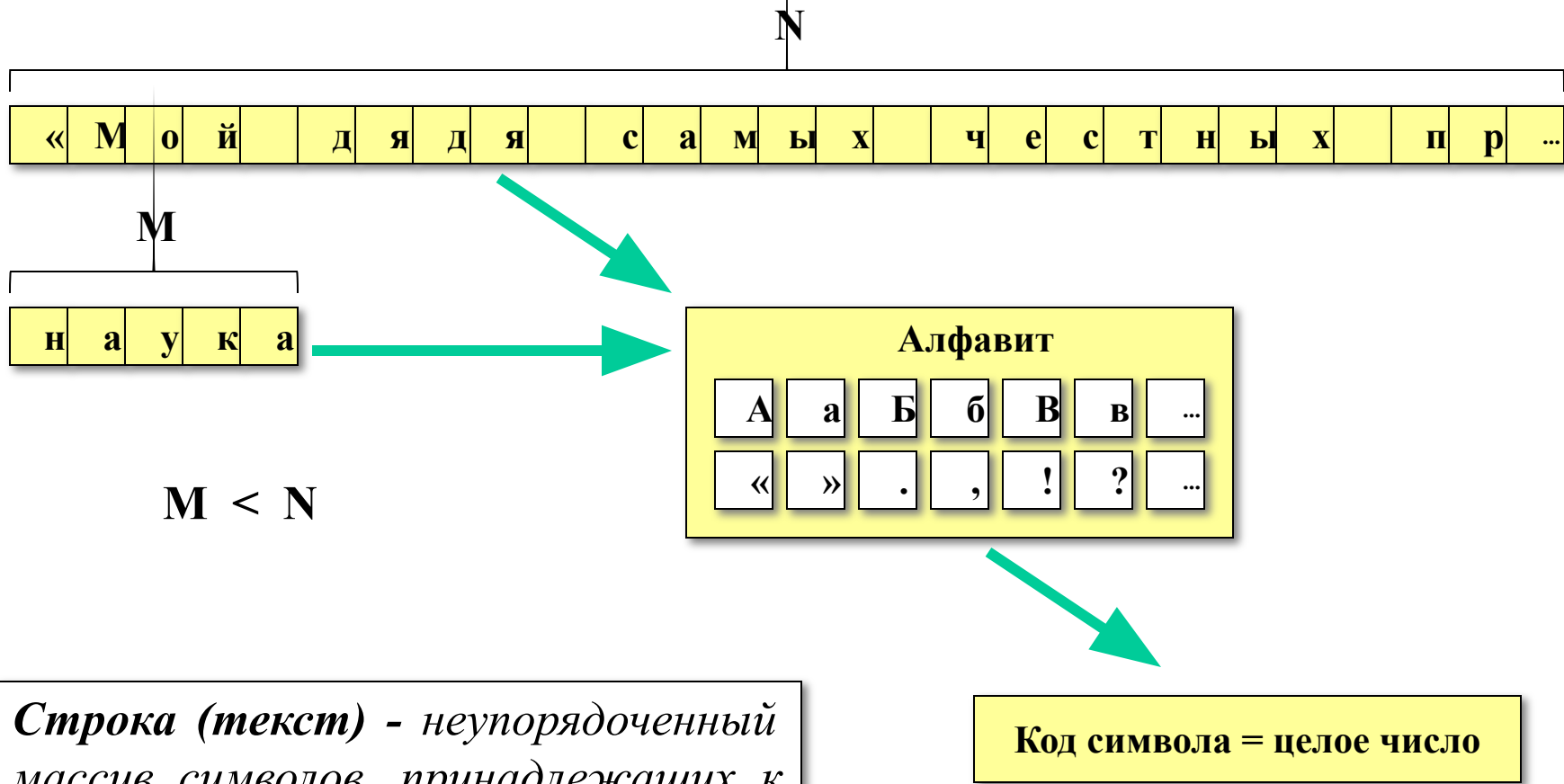
Поиск в строках



А. С. Пушкин
"ЕВГЕНИЙ ОНЕГИН"
ГЛАВА ПЕРВАЯ

«Мой дядя самых честных правил,
Когда не в шутку занемог,
Он уважать себя заставил
И лучше выдумать не мог.
Его пример другим **наука**;
Но, боже мой, какая скука
С больным сидеть и день и ночь,
Не отходя ни шагу прочь!
Какое низкое коварство
Полуживого забавлять,
Ему подушки поправлять,
Печально подносить лекарство,
Вздыхать и думать про себя:
Когда же черт возьмет тебя!»

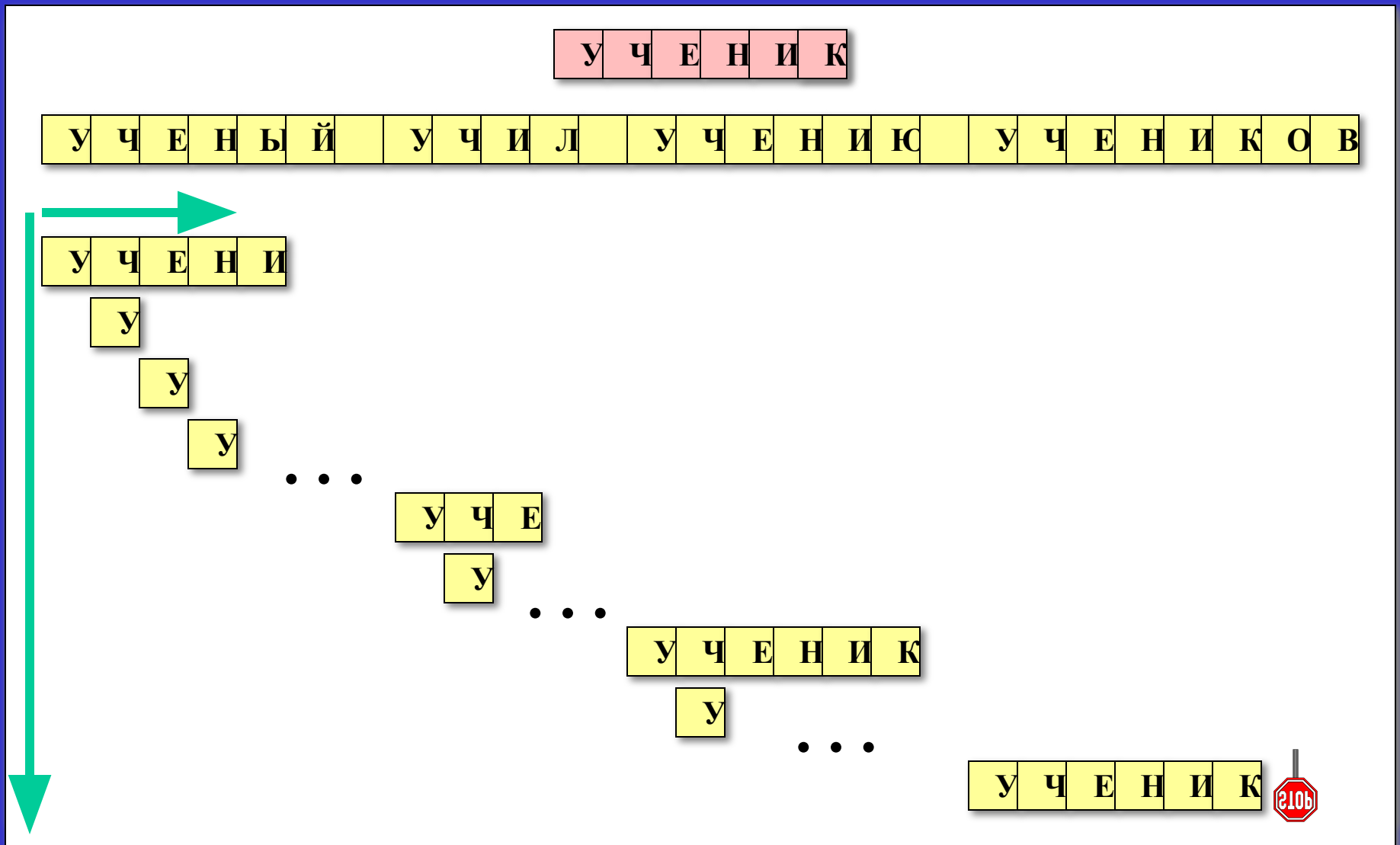
Задача поиска в строках



$$M < N$$

Строка (текст) - неупорядоченный массив символов, принадлежащих к некоторому алфавиту

Прямой поиск в строке



Прямой поиск в строке

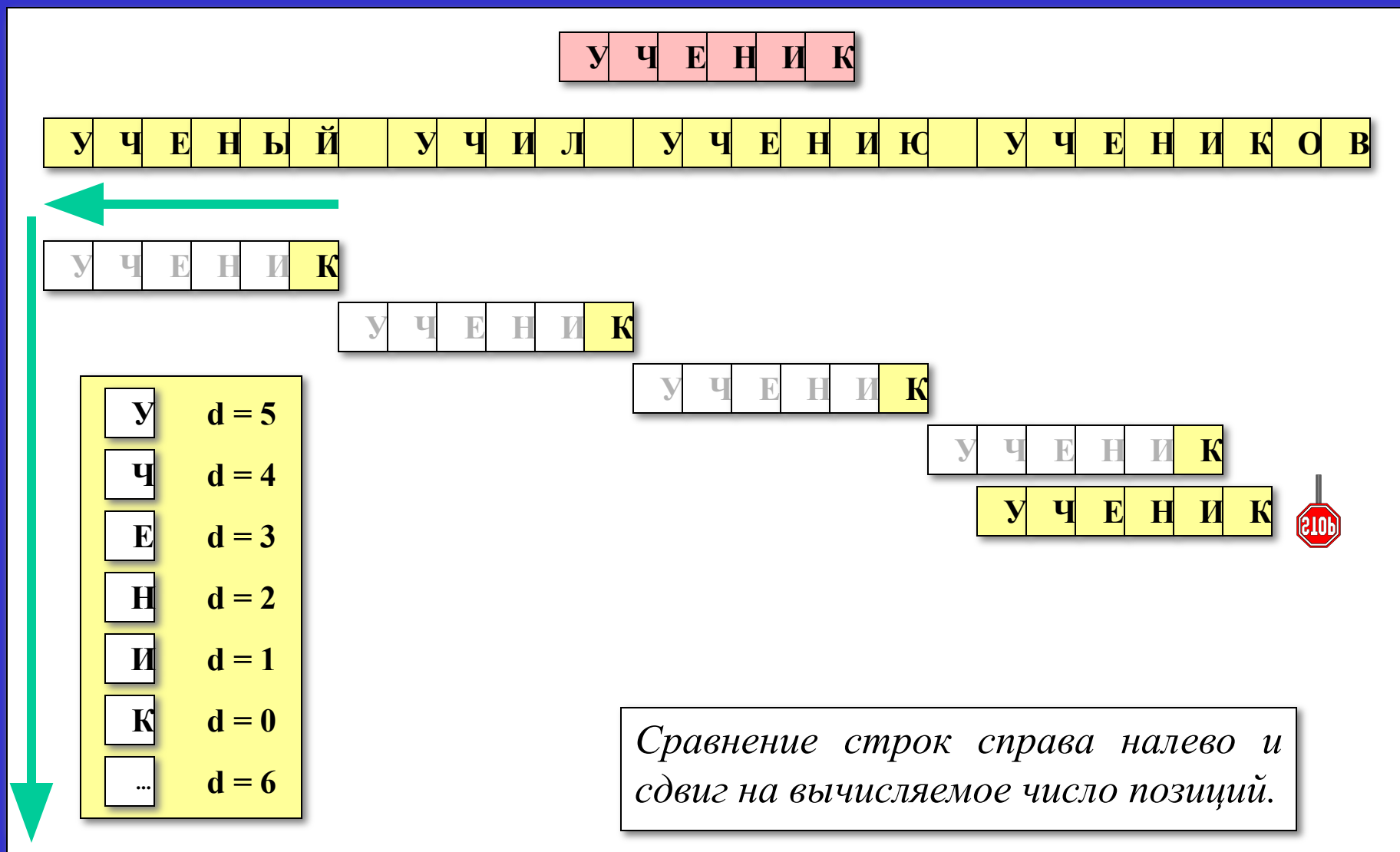
```
char* StringSearch(char* Text, char* Val)
{
    for(char* p = Text; *p; p++)
    {
        bool found = true;

        for(char* v = Val, *s = p; *v; v++, s++)
            if(!*s || *v != *s)
            {
                found = false;
                break;
            }

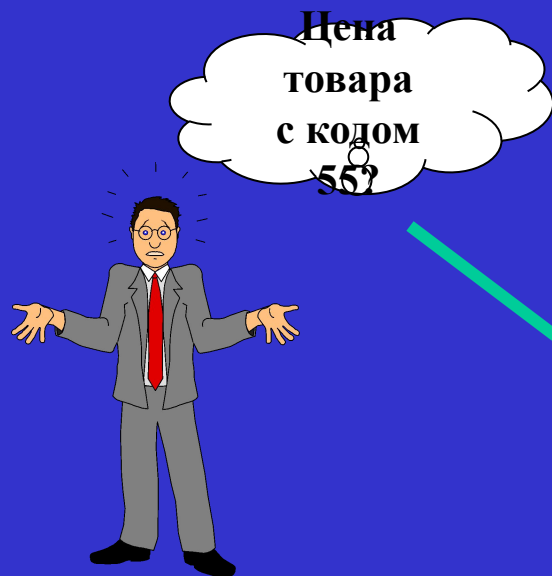
        if(found) return (p);
    }

    return (NULL);
}
```


Алгоритм Боуера и Мура



Поиск в файлах базы данных



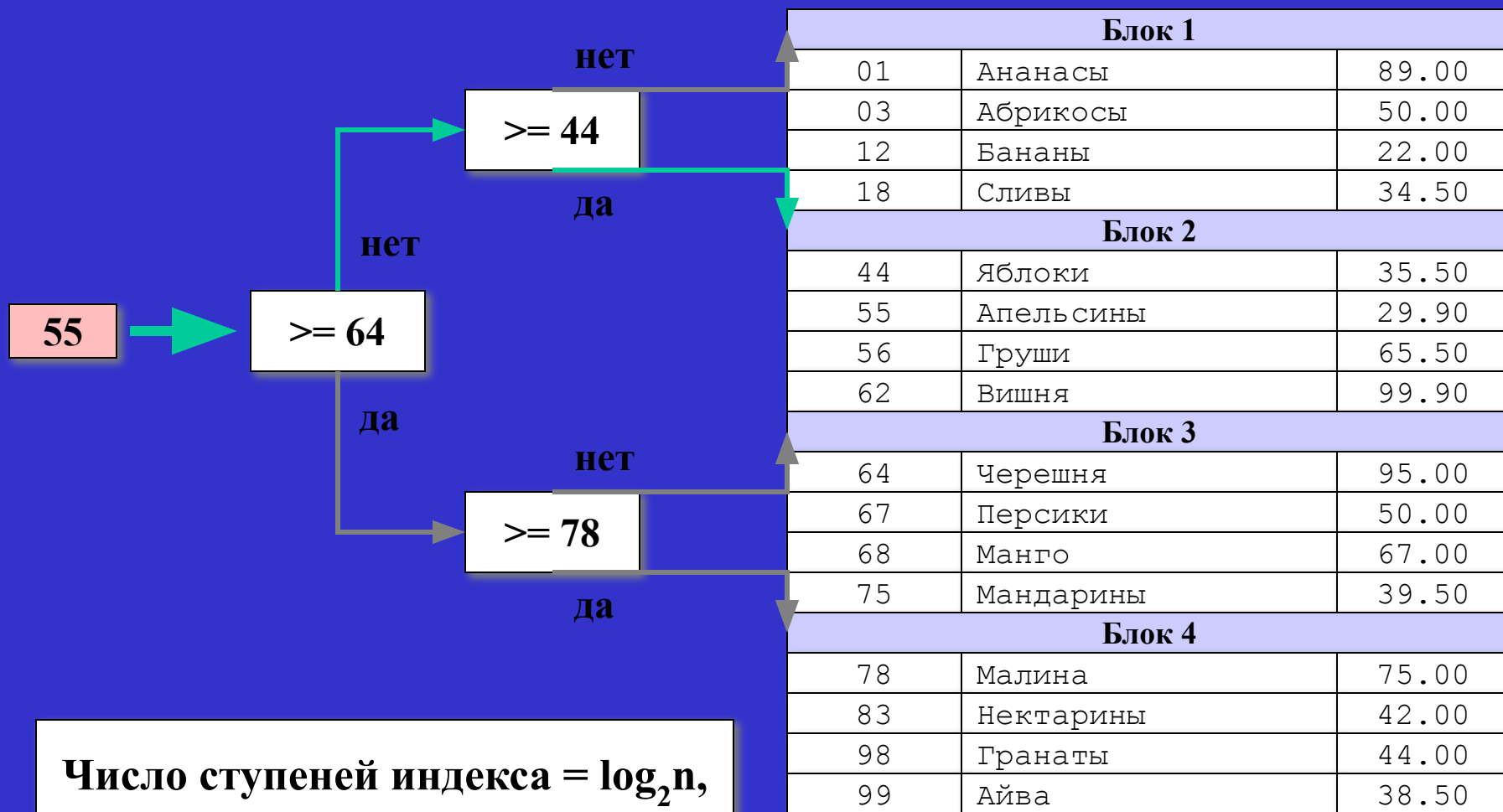
Код	Наименование	Цена
44	Яблоки	35.50
55	Апельсины	29.90
12	Бананы	22.00
...



Файлы базы данных размещаются на блочных устройствах (чтение и запись производится блоками фиксированной длины). Чтение блока - медленная операция.

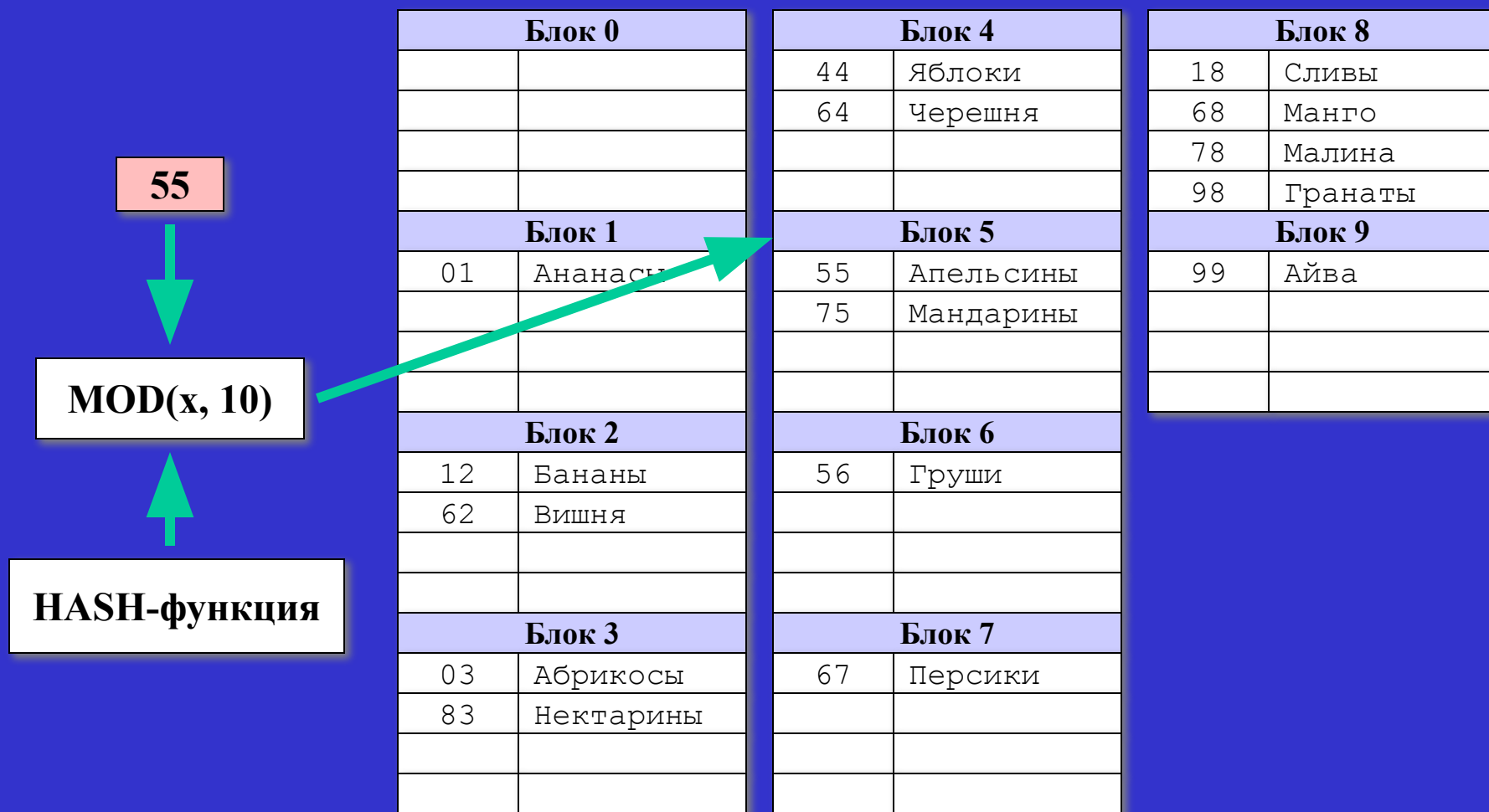


Индексно-последовательный метод доступа (ISAM)



**Число ступеней индекса = $\log_2 n$,
где n - число блоков**

Доступ по вычисляемому ключу (HASH)



Хэш-функция непосредственно преобразует значение ключа поиска в номер блока.