

Богатов Р.Н.

# Программирование на языке высокого уровня

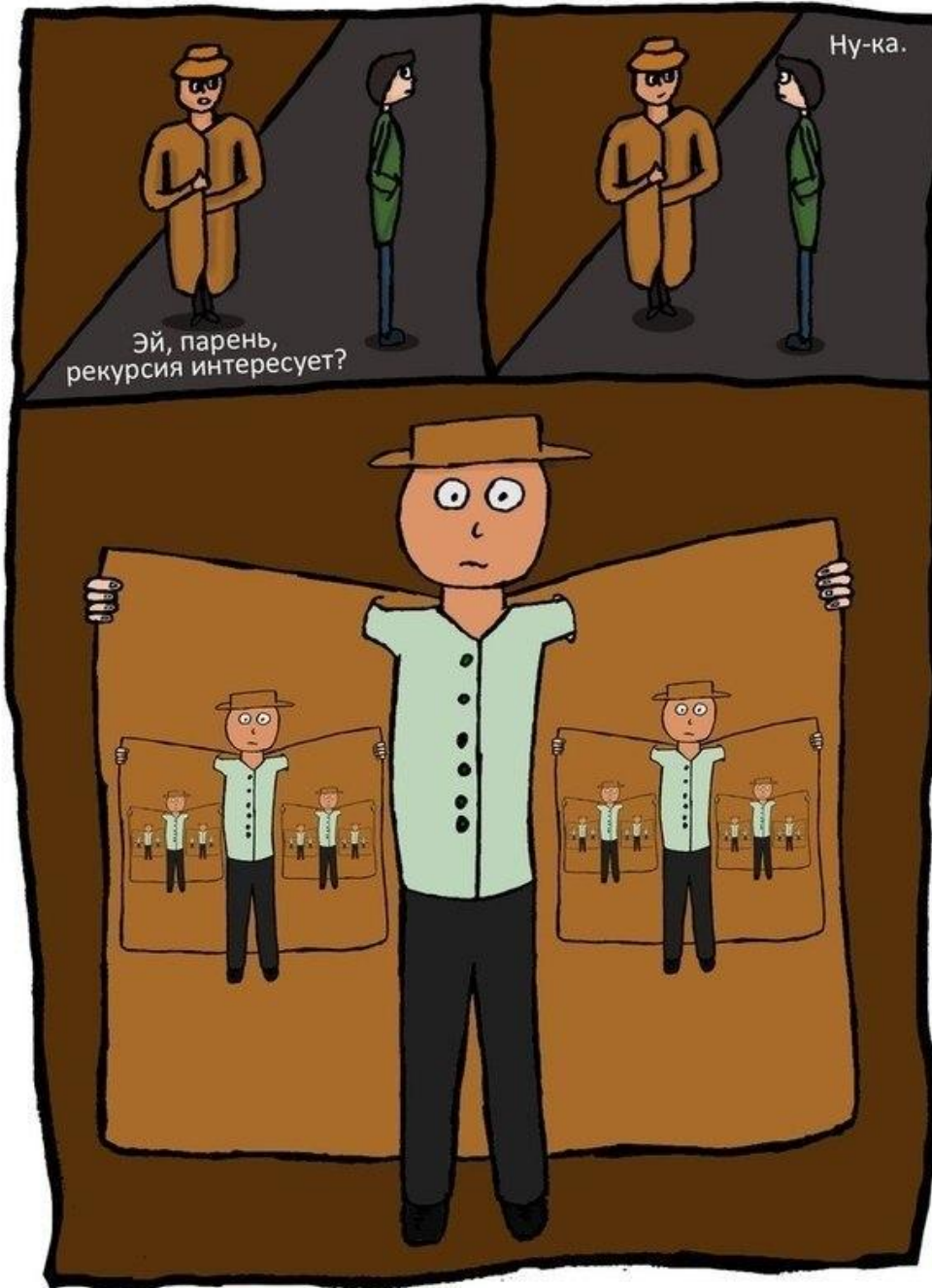
---

Лекция 13.

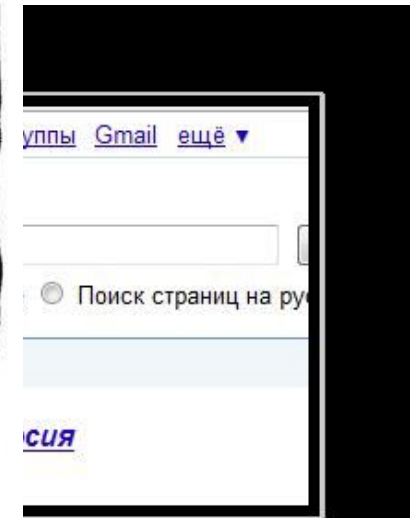
Рекурсия. Сложность алгоритмов

Кафедра АСОИУ ОмГТУ, 2013

# Чтобы понять рекурсию



ПОНЯТЬ



ПОНЯТЬ

ТО ЭТО ТАКОЕ



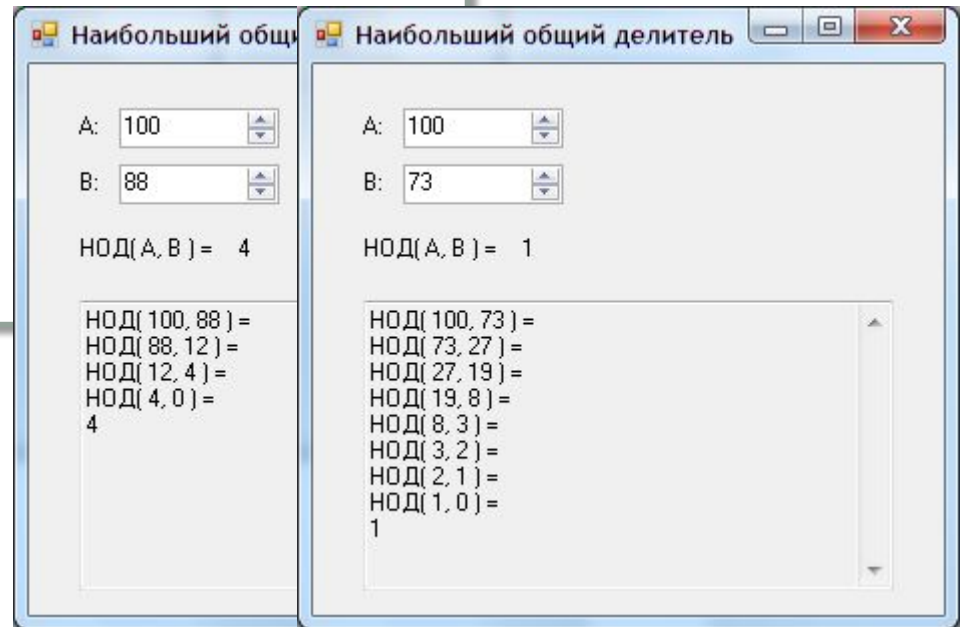
ТЕПЕРЬ И ТЫ

знаешь, что такое "рекурсия"

*ruKUD.ru*

# Алгоритм Евклида для вычисления НОД

```
{  
    ... = НОД( Math.Max(A, B), Math.Min(A, B) );  
}  
  
ulong НОД(ulong x, ulong y)  
{  
    if (y == 0)  
        return x;  
    else  
        return НОД(y, x % y);  
}
```



# Рекурсивный обход лабиринта

```
{
    if (!путь_из_(1, 1))
        MessageBox.Show("Этот лабиринт абсолютно точно непроходим!!!" );
}

bool путь_из_(int x, int y)
{
    m[x, y] = 1;

    if (x == N - 2 && y == N - 2) return true;

    if (m[x, y + 1] == 0)
        if (путь_из_(x, y + 1)) return true;
    if (m[x + 1, y] == 0)
        if (путь_из_(x + 1, y)) return true;
    if (m[x - 1, y] == 0)
        if (путь_из_(x - 1, y)) return true;
    if (m[x, y - 1] == 0)
        if (путь_из_(x, y - 1)) return true;

    m[x, y] = 0;
    return false;
}
```





## Мат. анализ: «O» большое

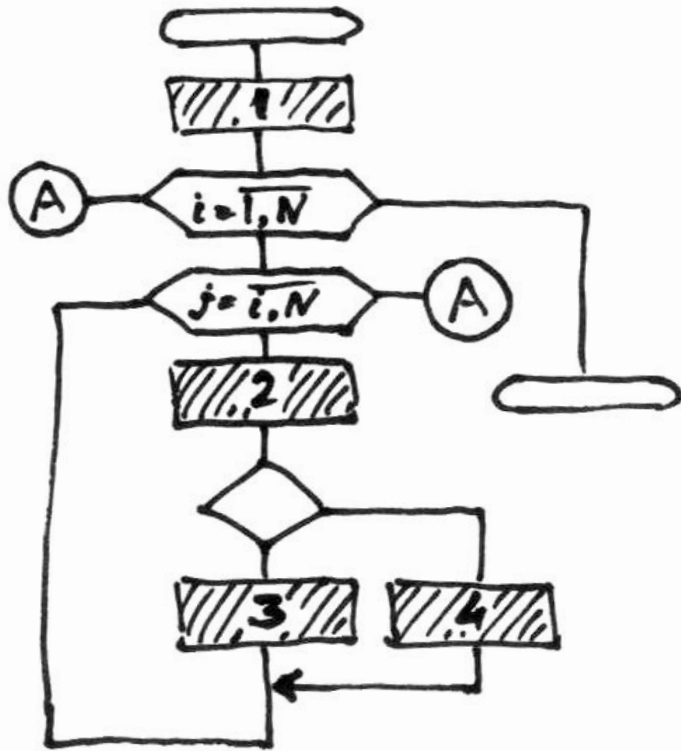
- Пишется:  $f(x) = O(g(x))$
- Читается:  $f$  является «O» большим от  $g$
- Формальное определение:  
 $f(x) = O(g(x))$ , если  $\exists x_0$  и  $c_0 \mid f(x) < c_0 g(x)$  при  $x > x_0$

## Сложность алгоритмов

- Говорят: «Сложность алгоритма есть  $O(N^2)$ »
- Значит: при больших  $N$  время работы алгоритма (или количество операций) не более чем  $c \cdot N^2$ , где  $c$  – положительная константа
- $N$  – обычно объём входных данных



## Оценка сложности алгоритма



$i$	Кол-во $j$
0	$N-1$
1	$N-2$
$\vdots$	$\vdots$
$N-1$	1

$$\Sigma = \frac{(1+N-1)(N-1)}{2} = \frac{N(N-1)}{2}$$

$$T_{\max} = t_1 + \frac{N(N-1)}{2} (t_2 + \max(t_3, t_4)) = \\ = t_1 + N^2 \frac{t_5}{2} - N \frac{t_5}{2} = O(N^2)$$