

# Основы анализа алгоритмов

0011 0010 1010 1101 0001 0100 1011

1 2  
4 5

Алгоритм - точное предписание,  
задающее процесс преобразования  
ИСХОДНЫХ ДАННЫХ в результаты.

- Свойства:
  - *Результативность*
  - *Конечность*
  - *Однозначность*
  - *Массовость*
  - *Детерминированность*
  - *Эффективность*



# Обозначения

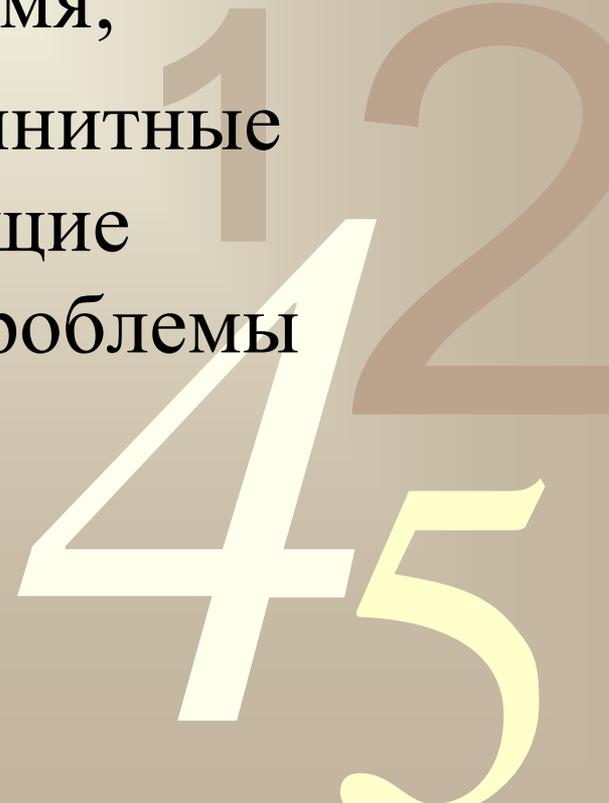
0011 0010 1010 1101 0001 0100 1011

- $M$  – набор машинных инструкций, составляющих программу;
- $S$  – память для стека и хранения промежуточных результатов;
- $d(n)$  – набор входных данных;
- $D(n) = \{d(n)\}$  – множество всевозможных наборов входных данных;
- $T = \{t\}$  – время выполнения программы;

# Допущения

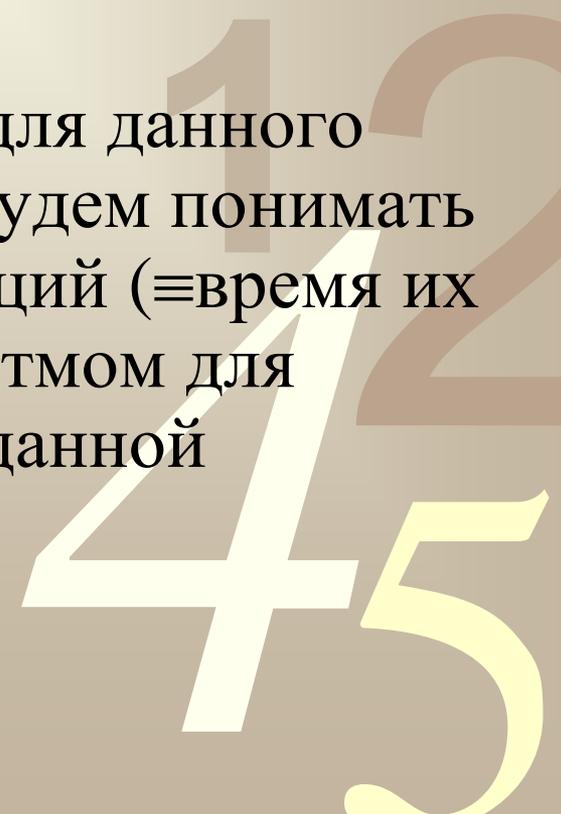
0011 0010 1010 1101 0001 0100 1011

- каждая машинная команда выполняется не более чем за фиксированное время;
- рассматриваем правильные и финитные алгоритмы, т.е. алгоритмы, дающие единственное решение общей проблемы



# В силу сделанных допущений

- Для любого начального набора данных алгоритм выполняет не более, чем конечное количество «элементарных» операций абстрактной машины.
- Под трудоемкостью  $F_d$  алгоритма для данного набора начальных данных –  $d(n)$ , будем понимать количество «элементарных» операций ( $\equiv$  время их выполнения), совершаемых алгоритмом для решения конкретной проблемы, в данной формальной системе.



# Время выполнения программы

$$t = c_1 * d(n) + c_2 * M + c_3 * S,$$

здесь  $c_i(d(n))$  – «веса» ресурсов

$T^{\wedge} = \max_{d \in D} T$  – наихудший случай

$T^{\vee} = \min_{d \in D} T$  – наилучший случай

$\check{T} = \Sigma t / |T|$  – средний случай



# Классификация алгоритмов по виду функции трудоёмкости

- *Параметрически-зависимые (порядко-независимые)*

$$F_d = f(n); T^{\wedge} = T^{\vee} = \check{T}$$

- *Порядко-зависимые*

$$F_d = f(d, n); T^{\vee}(d, n) \leq \check{T}(n) \leq T^{\wedge}(d, n)$$

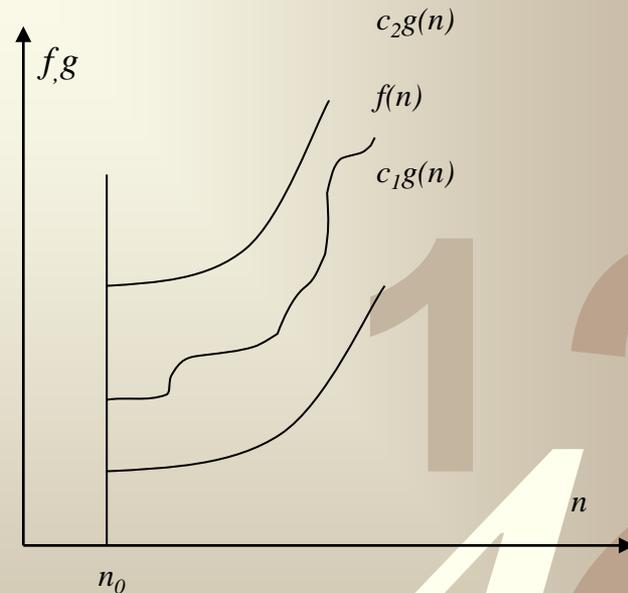
# Асимптотические оценки

- $\Theta$  (тета)

$f(n)$  и  $g(n)$ :  $\exists c_1, c_2, n_0: \forall n > n_0$

$$c_2 * g(n) \leq f(n) \leq c_1 * g(n) \rightarrow$$

$$f(n) = \Theta( g(n) )$$



Ex:  $F(x) = 4x^2 + \sin(x) = \Theta(x^2)$

# Асимптотические оценки

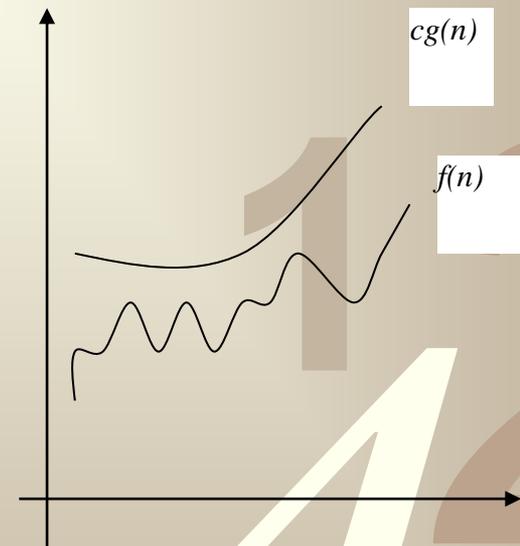
- O (O большое)

$f(n)$  и  $g(n)$ :  $\exists c, n_0: \forall n > n_0$

$c * g(n) \geq f(n) > 0 \rightarrow$

$f(n) = O( g(n) )$

Ex:  $F(x) = 4x^2 + x = O(x^2)$



45

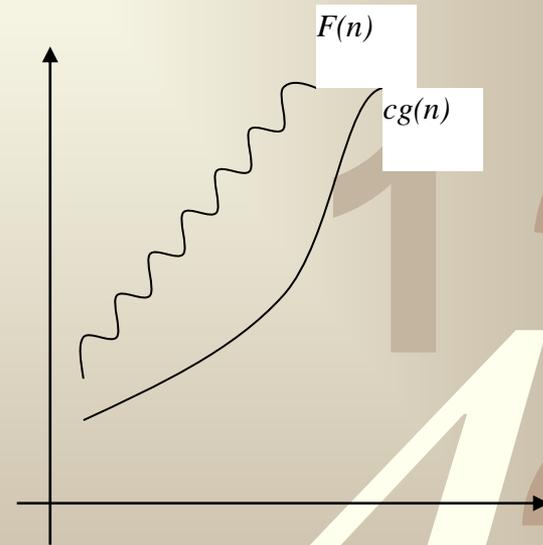
# Асимптотические оценки

- $\Omega$  (Омега)

$f(n)$  и  $g(n)$ :  $\exists c, n_0: \forall n > n_0$

$f(n) \leq c * g(n) \leq 0 \rightarrow$

$f(n) = \Omega( g(n) )$



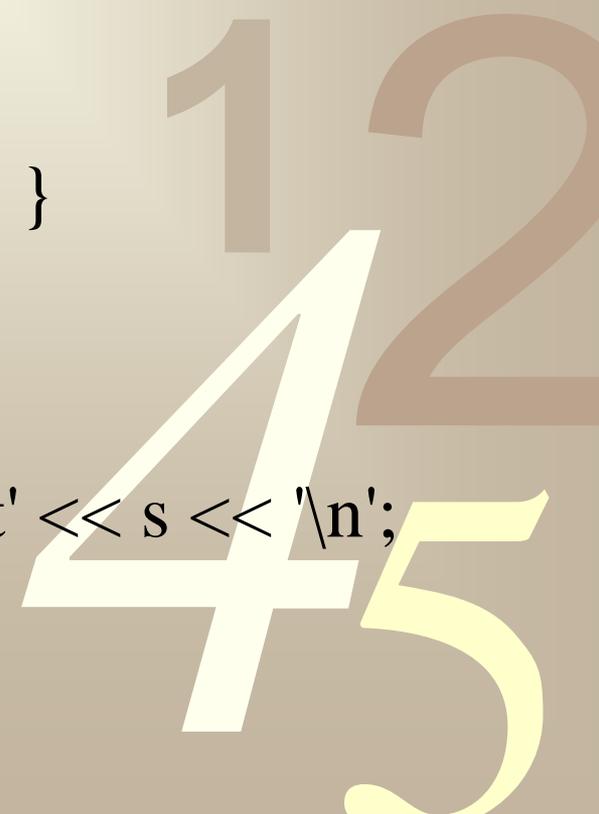
Ex:  $\Omega( x * \ln(x) )$  – класс функций, которые растут не медленнее, чем  $x * \ln(x)$

# Сравнение времен выполнения алгоритмов

	$n=10$	$n=10^3$	$n=10^6$
$\log_2 n$	0,2 сек	0,6 сек	1,2 сек
$n$	0,6 сек	1 мин	16,6 час
$n^2$	6 сек	16,6 час	1902 года
$2^n$	1 мин	$10^{295}$ лет	$10^{300000}$ лет

# Задача Иосифа

```
int main() { char *a; int n,m;
for(n=10; n<=10000; n*=10) for(m=10; m<=10000; m*=10)
{ int i, k=m, j=0; long s=0; a=new char[n];
for(i=0;i<n;i++) a[i]='\1';
for(i=1;i<n;i++,k=m) {
while(k) { if(a[j])k--; ++j%=n; s++; }
a[(j?j:n)-1]='\0'; }
i=0; while(!a[i++]);
cout << n << ' ' << m << '\t' << i << '\t' << s << '\n';
delete a; }
return 0; }
```





Project Explorer area with dropdown menus and file selection options.

Проект Классы Отладка ios1.cpp ios2.cpp ios3.cpp ios4.cpp ios1\_1.cpp tst.cpp

```

#include <iostream>
#include <time.h>
using namespace std;

int main() {
char *a; int n,m;
long t;
for(n=10; n<=10000; n*=10) for(m=10; m<=10000; m*=10) {
int i, k=m, j=0; long s=0; a=new char[n];
for(i=0;i<n;i++) a[i]='\1';
for(i=1;i<n;i++,k=m) { while(k) { if(a[j])k--; ++j%=n; s++; } a[(j?j:n)-1]='\0'; }
i=0; while(!a[i++]);
cout << n << ' ' << m << '\t' << i << '\t' << s << '\n';
delete a; }
cout << 0.001*clock() << " s" << endl;

return 0;
}

```

D:\A1\Lect2013\Exmp\6\ios1.exe

10	10	8	184	
10	100	6	1918	
10	1000	6	19278	
10	10000	4	192886	
100	10	26	3986	
100	100	22	41653	
100	1000	68	418497	
100	10000	21	4187144	
1000	10	63	61679	
1000	100	876	645192	
1000	1000	609	6481850	
1000	10000	4	64851200	
10000	10	9143	835312	
10000	100	9332	8742515	
10000	1000	5981	87828197	
10000	10000	2019	878714110	
			25.194 s	

Компилятор Ресурсы Журнал компиляции Отладка Результаты поиска Закреть

Отладка Отслеживать Вывод

Следующий шаг   
 Продолжить выполнение   
 Отладка   
 Добавить в наблюдаемые  
 Шаг внутрь   
 Выполнить до курсора   
 Остановить выполнение   
 Удалить объект наблюдения

# Задача Иосифа

- Количество сравнений

– на полном наборе:  $m * n / n$

– на последнем шаге:  $n * m / 2$

– на  $i$ -ом шаге:  $n * m / (n - i)$

• Всего:  $m + \dots + n * m / (n - i) + \dots + n * m / 2 =$   
 $= m * n * (1/n + \dots + 1/(n - i) + \dots + 1/2) = m * n (\ln(n) + 0,57 - 1)$

```
for(i=1;i<n;i++,k=m) {  
    while(k) { if(a[j])k--; ++j%=n; s++; }  
    a[(j?j:n)-1]='\0'; }
```

# Задача Йосиф

0011 0010 1010 1101 0001 0100 1011

```
D:\A1\Lect2013\Exmp\6\ios1.exe
10 10 8 184
10 100 6 1918
10 1000 6 19278
10 10000 4 192886
100 10 26 3986
100 100 22 41653
100 1000 68 418497
100 10000 21 4187144
1000 10 63 61679
1000 100 876 645192
1000 1000 609 6481850
1000 10000 4 64851200
10000 10 9143 835312
10000 100 9332 8742515
10000 1000 5981 87828197
10000 10000 2019 878714110
25.194 s
```

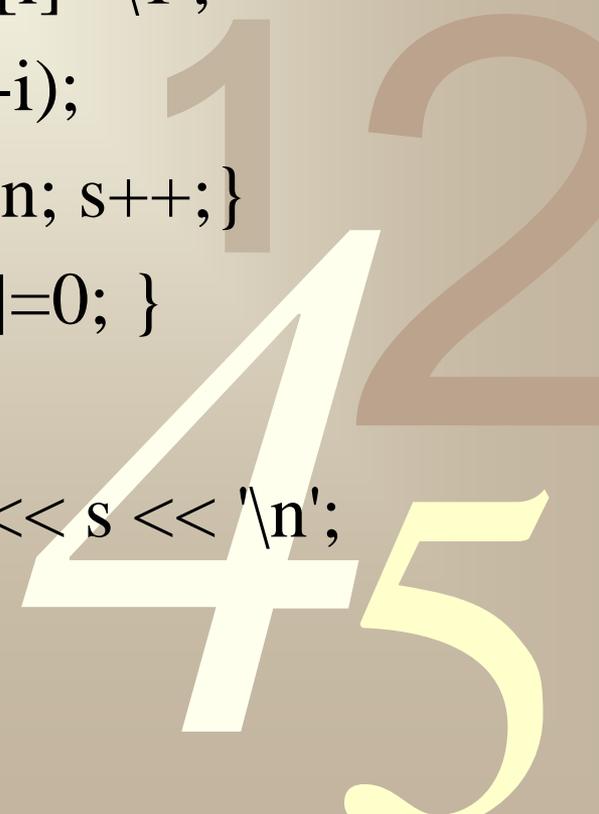
m \ n		m			
		~ Ln(n) -0,43	10	100	1000
10	1,9	184	1 918	19 278	192 886
100	4,2	3 984	41 953	418 497	4 187 144
1000	6,5	61 679	645 192	6 481 850	64 851 200
10000	8,8	835 312	8 742 515	87 828 197	878 714 110

$\sim n * m * \ln(n)$

25,194 сек

# Задача Иосифа (1)

```
int main() { char *a; int n,m,k;  
for(n=10; n<=10000; n*=10)  
    for(m=10; m<=10000; m*=10) { int i, j=0; long s=0;  
        a=new char[n]; for(i=0;i<n;i++) a[i]='\1';  
        for(i=0;i<n-1;++i) { k=(m-1)%(n-i);  
            while(k) { if(a[j]) k--; ++j%=n; s++;}  
            while(!a[j]) ++j%=n,s++; a[j]=0; }  
        i=0; while(!a[i++]);  
        cout << n << ' ' << m << '\t' << i << '\t' << s << '\n';  
        delete a; }  
return 0; }
```





Project Explorer showing files: ios1.cpp, ios2.cpp, ios3.cpp, ios4.cpp, ios1\_1.cpp, tst.cpp

Debugger window: Could not watch this variable

```

#include <iostream>
#include <time.h>
using namespace std;

int main() {
char *a; int n,m;
long t;
for(n=10; n<=10000; n*=10) for(m=10; m<=10000; m*=10) {
//      n=1000; m=2000; {
int i, k, j=0; long s=0; a=new char[n];
for(i=0;i<n;i++) a[i]='\1';
for(i=0;i<n-1;++i) { k=(m-1)%(n-i); while(k) { if(a[j]) k--; ++j%=n; s++;
while(!a[j]) ++j%=n,s++;
a[j]=0;
}

i=0; while(!a[i++]);
cout << n << ' ' << m << '\t' << i << '\t' << s << '\n';

delete a; }
cout << 0.001*clock() << endl;

return 0;
}

```

D:\A1\Lect2013\Exmp\6\ios1\_1.exe

```

10 10 8 43
10 100 6 47
10 1000 6 57
10 10000 4 55
100 10 26 2585
100 100 22 4252
100 1000 68 5296
100 10000 21 4943
1000 10 63 47678
1000 100 876 271191
1000 1000 609 427849
1000 10000 4 490199
10000 10 9143 695311
10000 100 9332 5002514
10000 1000 5981 27288196
10000 10000 2019 42274109
2.137

```

Отладка | Отслеживать | Вывод

Следующий шаг   
 Продолжить выполнение   
 Отладка   
 Добавить в наблюдаемые  
 Шаг внутрь   
 Выполнить до курсора   
 Остановить выполнение   
 Удалить объект наблюдения

# Задача Иосиф

0011 0010 1010 1101 0001 0100 1011

```
D:\A1\Lect2013\Exmp\6\ios1_1.exe
10 10 8 43
10 100 6 47
10 1000 6 57
10 10000 4 55
100 10 26 2585
100 100 22 4252
100 1000 68 5296
100 10000 21 4943
1000 10 63 47678
1000 100 876 271191
1000 1000 609 427849
1000 10000 4 490199
10000 10 9143 695311
10000 100 9332 5002514
10000 1000 5981 27288196
10000 10000 2019 42274109
2.137
```

N \ M	10	100	1 000	10 000
10	43	47	57	55
100	2 585	4 252	5 296	4 943
1 000	47 678	271 191	427 849	490 199
10 000	695 311	5 002 514	27 288 196	42 274 109

2,137 сек

# Задача Иосифа (2)

```
int main() { int *a; int n,m;
    for(n=10; n<=10000; n*=10)
        for(m=10; m<=10000; m*=10) { long s=0; int j=0, i;
            a = new int[n]; for(i=0;i<n;i++) a[i]=i+1;
            for(i=0;i<n-1;i++) { j=(j+m-1)%(n-i);
                for(int k=j;k<n-1-i;k++) a[k]=a[k+1],s++; }
            cout << n << ' ' << m << '\t' << a[0] << '\t' << s << '\n';
            delete a; }
    return 0; }
```

45



Создать Вставить Переключить Перейти

Проект Классы Отладка

ios1.cpp ios2.cpp ios3.cpp ios4.cpp ios\_1.cpp tst.cpp

Could not watch this variable

```
#include <iostream>
using namespace std;

int main() {
int *a; int n,m;
for(n=10; n<=10000; n*=10) for(m=10; m<=10000; m*=10) {
long s=0; int j=0; int i;
a = new int[n]; for(i=0;i<n;i++) a[i]=i+1;
for(i=0;i<n-1;i++) {
j=(j+m-1)%(n-i);
for(int k=j;k<n-1-i;k++) a[k]=a[k+1],s++;
}
cout << n << ' ' << m << '\t' << a[0] << '\t' << s << '\n';
delete a; }
cout << 0.001*clock() << " s" << endl;

return 0;
}
```

D:\A1\Lect2013\Exmp\6\ios2.exe

```
10 10 8 26
10 100 6 24
10 1000 6 20
10 10000 4 18
100 10 26 2442
100 100 22 2838
100 1000 68 2446
100 10000 21 2606
1000 10 63 262374
1000 100 876 249264
1000 1000 609 262062
1000 10000 4 250950
10000 10 9143 26292352
10000 100 9332 25106354
10000 1000 5981 25053310
10000 10000 2019 25270108
0.982 s
```

Компилятор Ресурсы Журнал компиляции Отладка Результаты поиска Закрыть

Отладка Отслеживать Вывод

Следующий шаг Продолжить выполнение Отладка Добавить в наблюдаемые  
Шаг внутрь Выполнить до курсора Остановить выполнение Удалить объект наблюдения

# Задача Иосифа (2)

0011 0010 1010 1101 0001 0100 1011

- Среднее количество перестановок

$$p_i(n) = \frac{(n - i - 1) + \dots + 1}{n - i} = \frac{n - i - 1}{2}$$

- Полное количество перестановок

$$P(n) = \sum p_i = \frac{n-1}{2} + \frac{n-2}{2} + \dots + \frac{1}{2} = \frac{1}{2}(n-1) \frac{n-1+1}{2} = \frac{n(n-1)}{4}$$

# Задача Иосифа (2)

0011 0010 1010 1101 0001 0100 1011

N \ M	10	100	1 000	10 000
10	26	24	20	18
100	2 442	2 838	2 446	2 606
1 000	262 374	249 264	262 062	250 950
10 000	26 292 352	25 106 354	25 053 310	25 270 108

$\sim n*n/4$

0,982 сек

# Задача Иосифа (3)

```
int main() { int n,m;
for(n=10;n<=10000;n*=10)
  for(m=10;m<=10000;m*=10) {
    int *a = new int[n],i,j=0,k=(m-1)%n; long s=0;
    for(i=0;i<n-1;i++) a[i]=i+1; a[n-1]=0;
    for(i=0;i<n-1;i++,k=(m-1)%(n-i)) {
      while(k--) j=a[j],s++; a[j]=a[a[j]]; }
    std::cout << n << ' ' << m << '\t' << j << '\t' << s << '\n'; }
  return 0;
}
```



Проект Классы Отладка ios1.cpp ios2.cpp ios3.cpp ios4.cpp ios1\_1.cpp tst.cpp

Could not watch this variable

```
#include <iostream>

int main()
{   int n,m;
    for(n=10;n<=10000;n*=10) for(m=10;m<=10000;m*=10) {
        int *a = new int[n],i,j=0,k=(m-1)%n; long s=0;
        for(i=0;i<n-1;i++) a[i]=i+1; a[n-1]=0;
        for(i=0;i<n-1;i++,k=(m-1)%(n-i)) {
            while(k--) j=a[j],s++; a[j]=a[a[j]]; }
        std::cout << n << ' ' << m << '\t' << j << '\t' << s << '\n'; }
    std::cout << 0.001*clock() << " s\n";

    return 0;
}
```

```
D:\A1\Lect2013\Exmp\6\ios3.exe
10 10 8      21
10 100 6     24
10 1000 6    32
10 10000 4    30
100 10 26   831
100 100 22  1818
100 1000 68  2575
100 10000 21  2451
1000 10 63  8931
1000 100 876  90918
1000 1000 609 178259
1000 10000 4  241316
10000 10 9143 89931
10000 100 9332 981918
10000 1000 5981 9169259
10000 10000 2019 17758197
0.327 s
```

Компилятор Ресурсы Журнал компиляции Отладка Результаты поиска Закрыть

Отладка Отслеживать Вывод

Следующий шаг Продолжить выполнение Отладка Добавить в наблюдаемые  
Шаг внутрь Выполнить до курсора Остановить выполнение Удалить объект наблюдения

# Задача Иосифа (3)

0011 0010 1010 1101 0001 0100 1011

N \ M	10	100	1 000	10 000
10	21	24	32	30
100	831	1 818	2 575	2 451
1 000	8 931	90 918	178 259	241 316
10 000	89 931	981 918	9 169 259	17 758 197

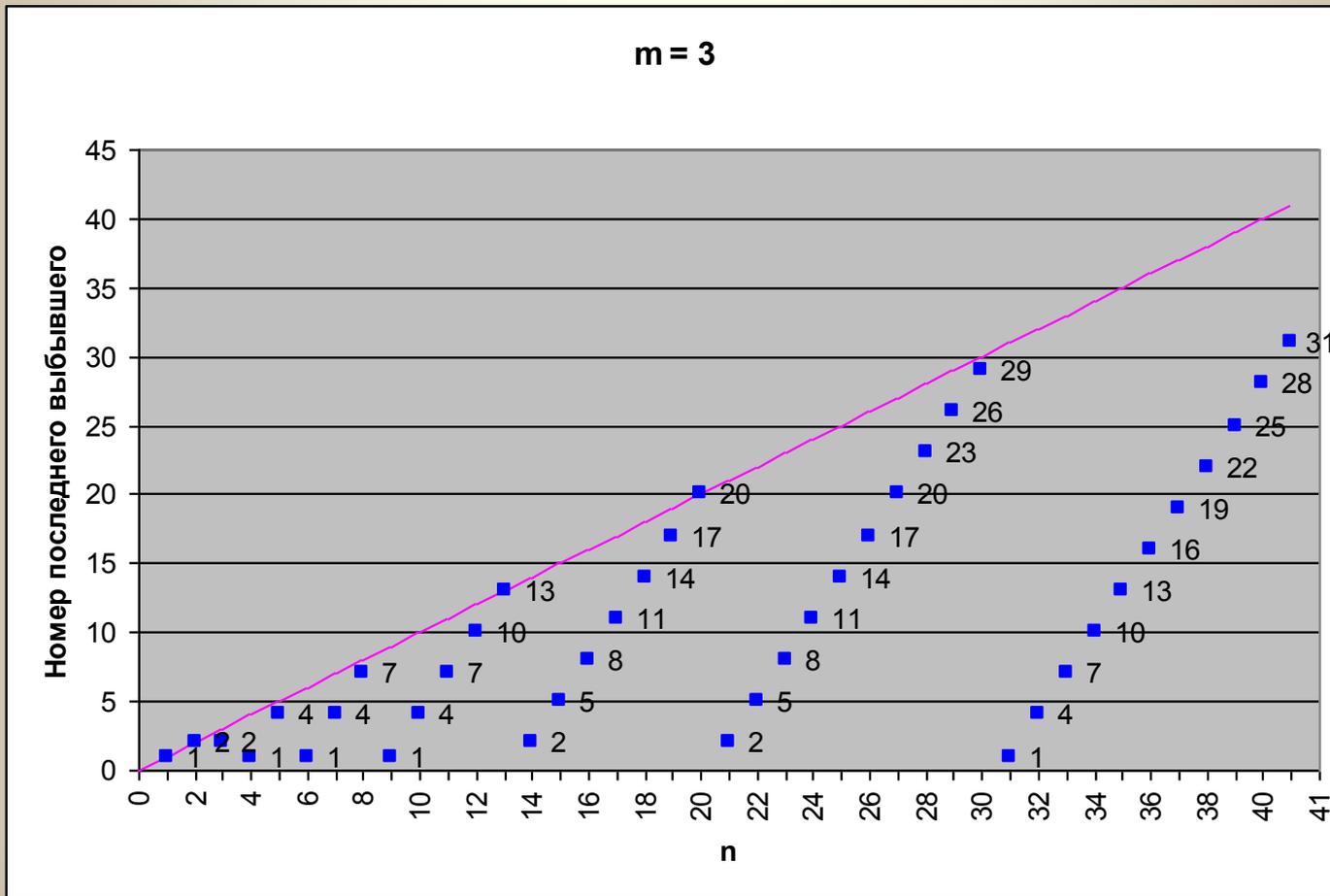
$M \geq N: \sim n * n / 4$

$M < N: \sim n * m - \frac{3}{4} * m * m$

0,327 c

# Задача Иосифа (4)

m = 3



$$F(1)=1; F(n)=(F(n-1) + m)_{\text{mod } n}$$

# Задача Иосифа (4)

```
int m;
```

```
int f(int n) {
```

```
    if(n>1) {int p=(f(n-1)+m)%n; return p?p:n; }
```

```
    else return 1; }
```

```
int main() { int n;
```

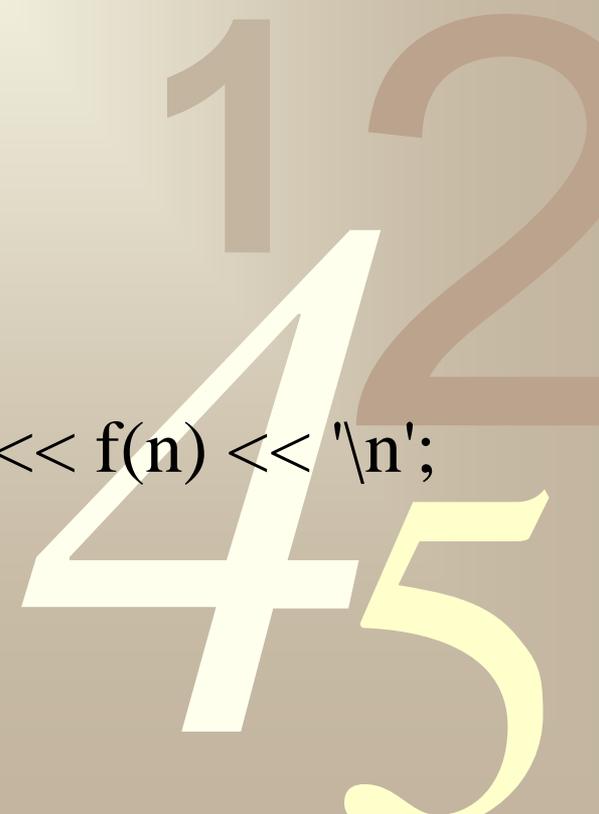
```
    for(n=10;n<=10000;n*=10)
```

```
        for(m=10;m<=10000;m*=10)
```

```
            cout << n << ' ' << m << '\t' << f(n) << '\n';
```

```
        cout << clock() << " ms\n";
```

```
    return 0; }
```



Создать Вставить Переключить Перейти

Проект Классы Отладка ios4.cpp

```

#include <iostream>
using namespace std;

int m;
int f(int n) {
    if(n>1) {int p=(f(n-1)+m)%n; return p?p:n; }
    else return 1; }

int main() {    int n;
    for(n=10;n<=10000;n*=10) for(m=10;m<=10000;m*=10)
        cout << n << ' ' << m << '\t' << f(n) << '\n';
    cout << clock() << " ms\n";

    return 0;
}

```

D:\A1\Lect2013\Exmp\6\ios4.exe

```

10 10 8
10 100 6
10 1000 6
10 10000 4
100 10 26
100 100 22
100 1000 68
100 10000 21
1000 10 63
1000 100 876
1000 1000 609
1000 10000 4
10000 10 9143
10000 100 9332
10000 1000 5981
10000 10000 2019
31 ms

```

Компилятор Ресурсы Журнал компиляции Отладка Результаты поиска Закрыть

Отладка Отслеживать Вывод