

Информатика

Введение. Основные определения информатики и информации

Максутов Артем Артурович,
Ассистент кафедры №12 «Компьютерные системы и технологии»
aamaksutov@mephi.ru

Москва, 2017



Информатика - наука, изучающая способы автоматизированного создания, хранения, обработки, использования, передачи и защиты информации.

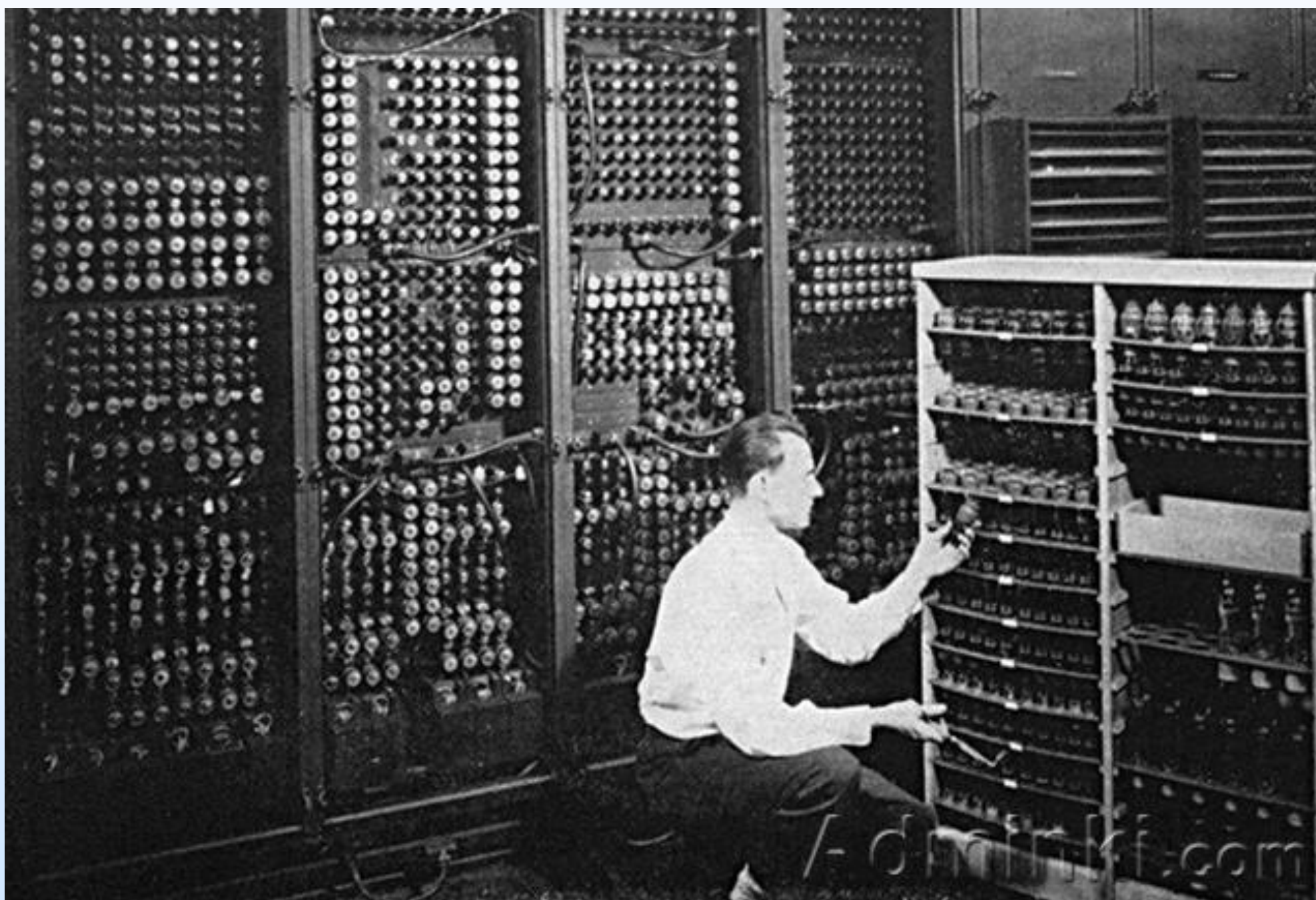
Информация - набор символов, графических образов или звуковых сигналов, несущих определенную смысловую нагрузку.



ЭВМ

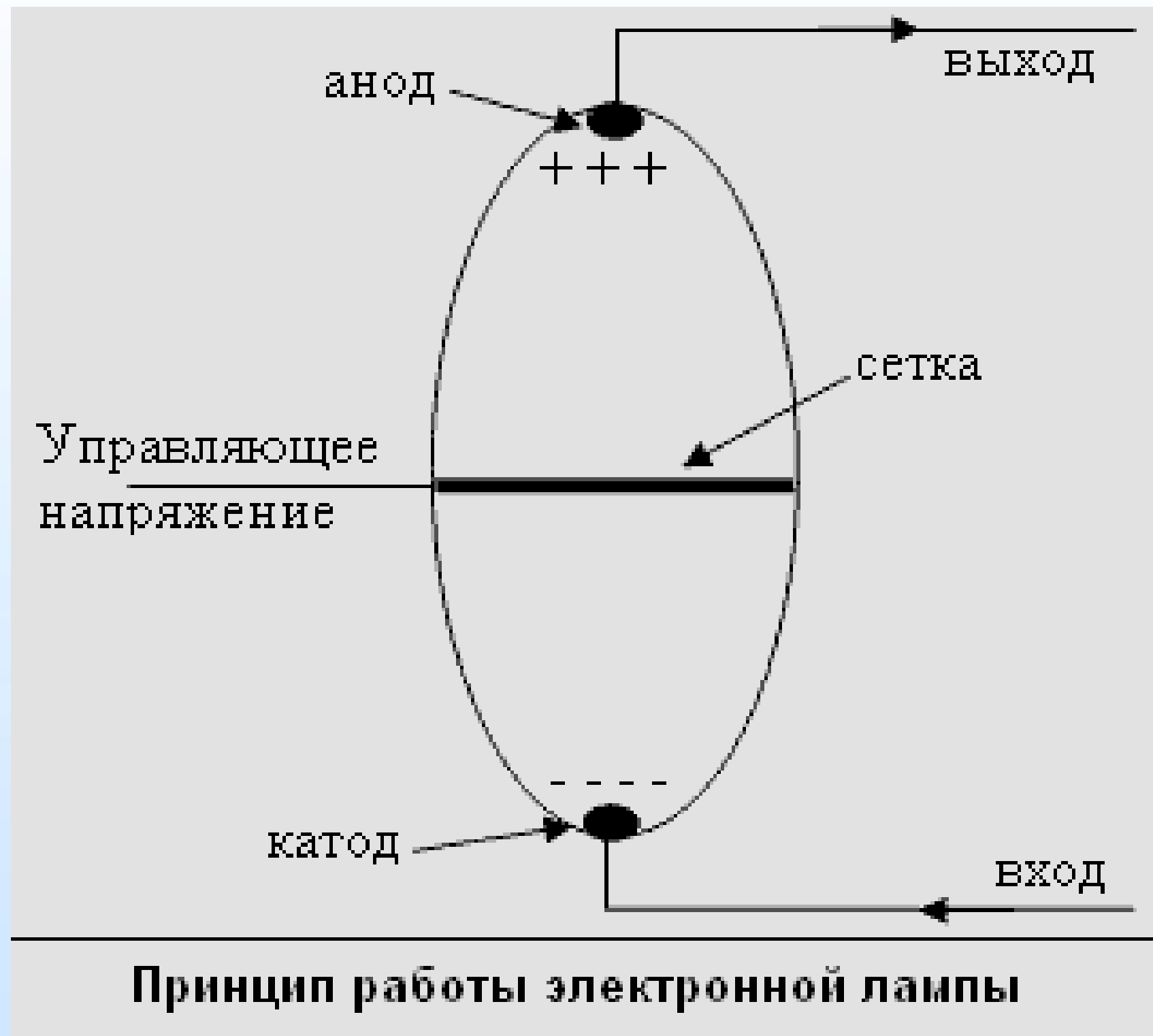
- Электронно-вычислительная машина (ЭВМ) или компьютер (англ. Computer - вычислитель)- устройство для автоматизированной обработки информации

Ламповые дни





Это вообще как?

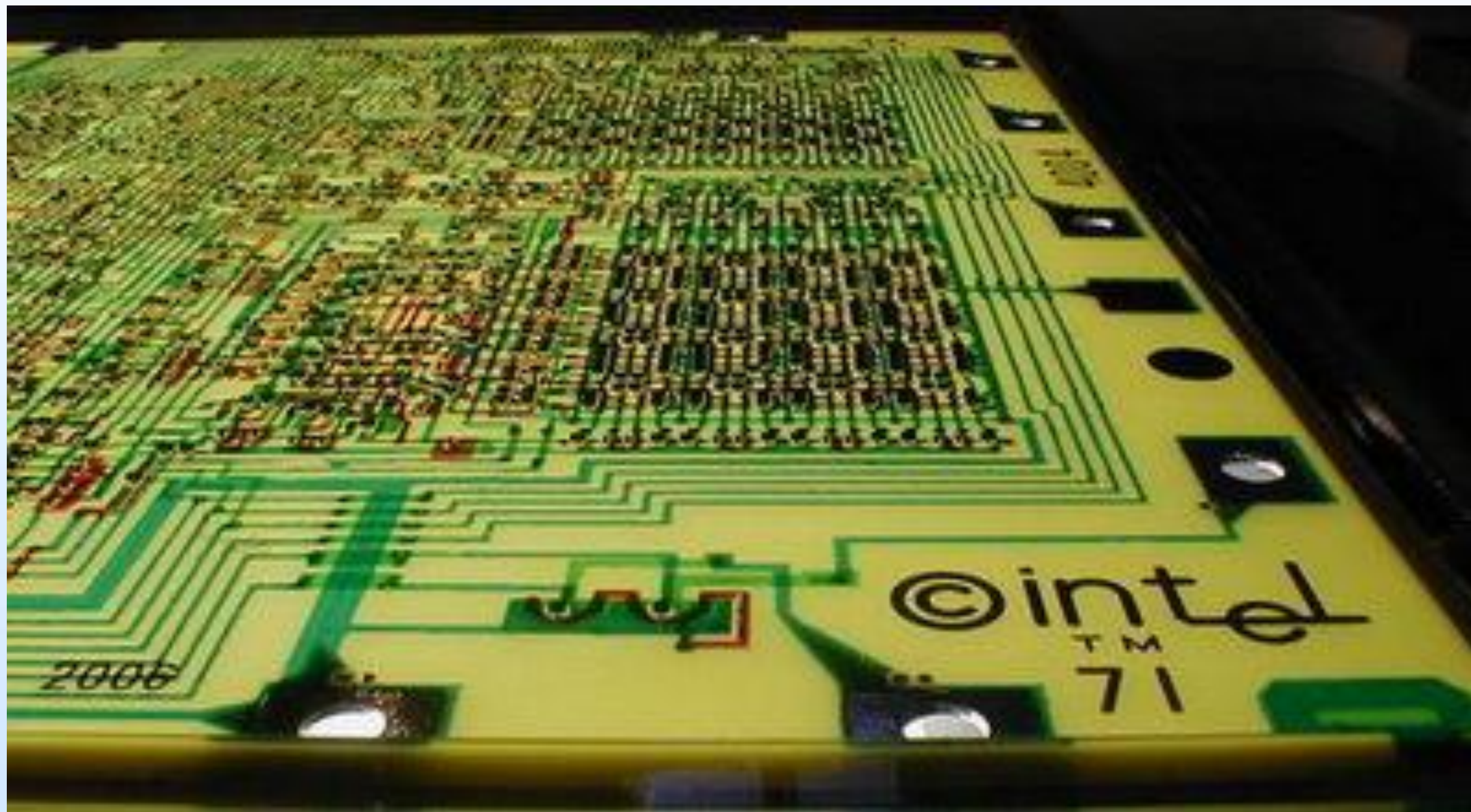


Транзисторы



Микросхемы



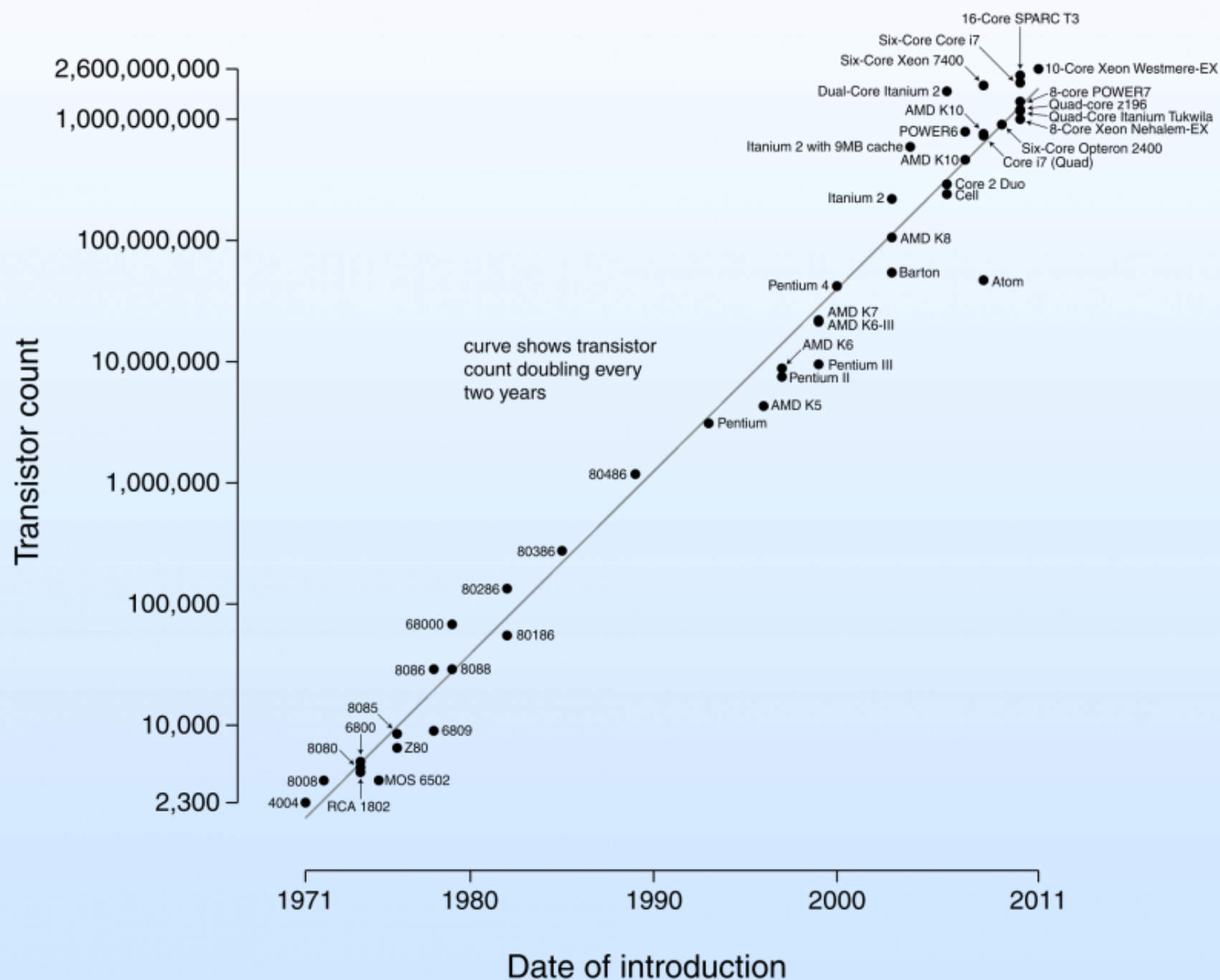


Закон Мура

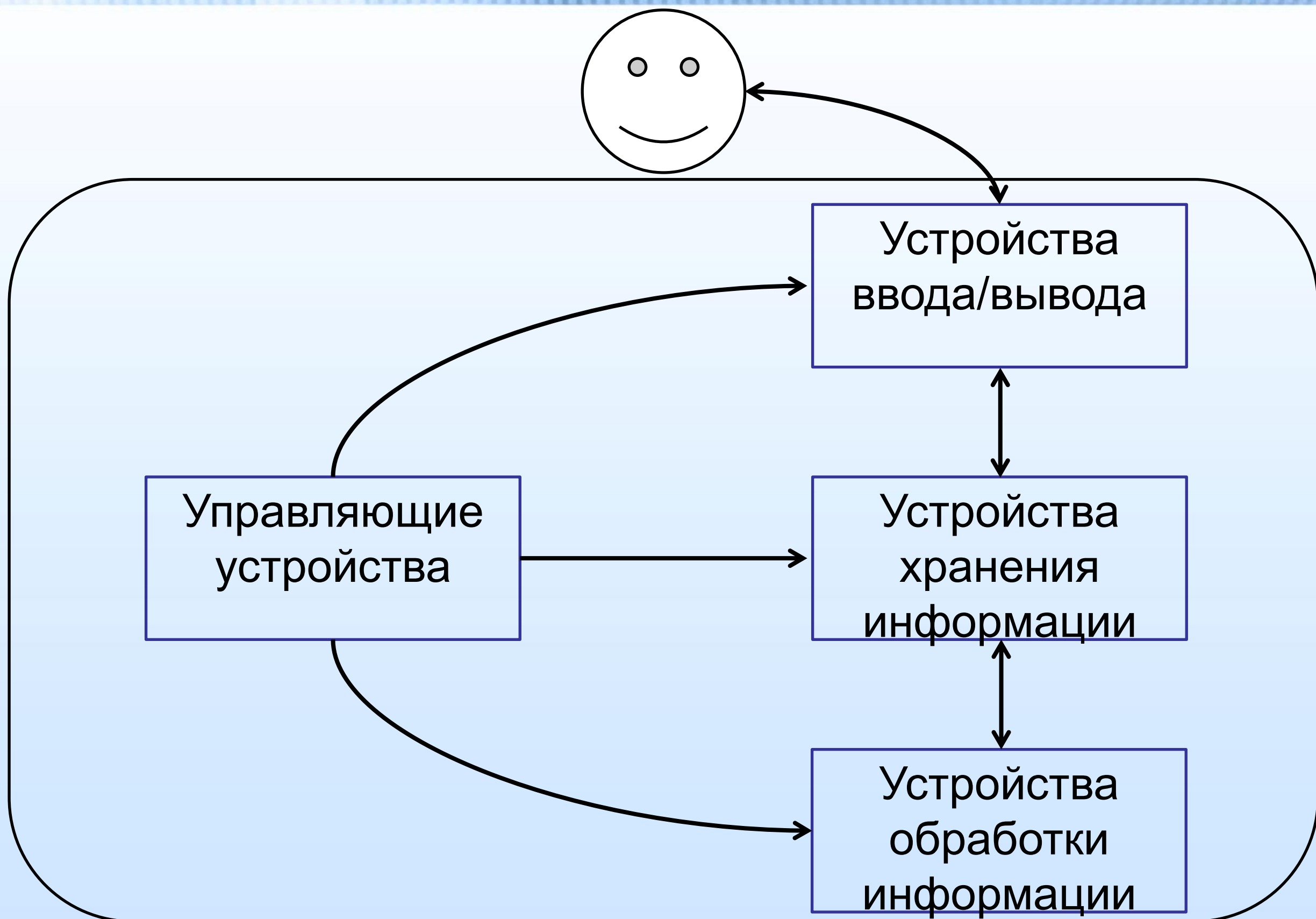
- Количество транзисторов, размещаемых на кристалле интегральной схемы, удваивается каждые 24 месяца.

Что это значит?

Microprocessor Transistor Counts 1971-2011 & Moore's Law



Компьютер – это просто



Спасибо за внимание!

aamaksutov@mephi.ru



ИНФОРМАТИКА

Тема: Данные: мир глазами компьютера

Максутов Артем Артурович,
Ассистент кафедры 12

Москва, 2017

Мои примеры

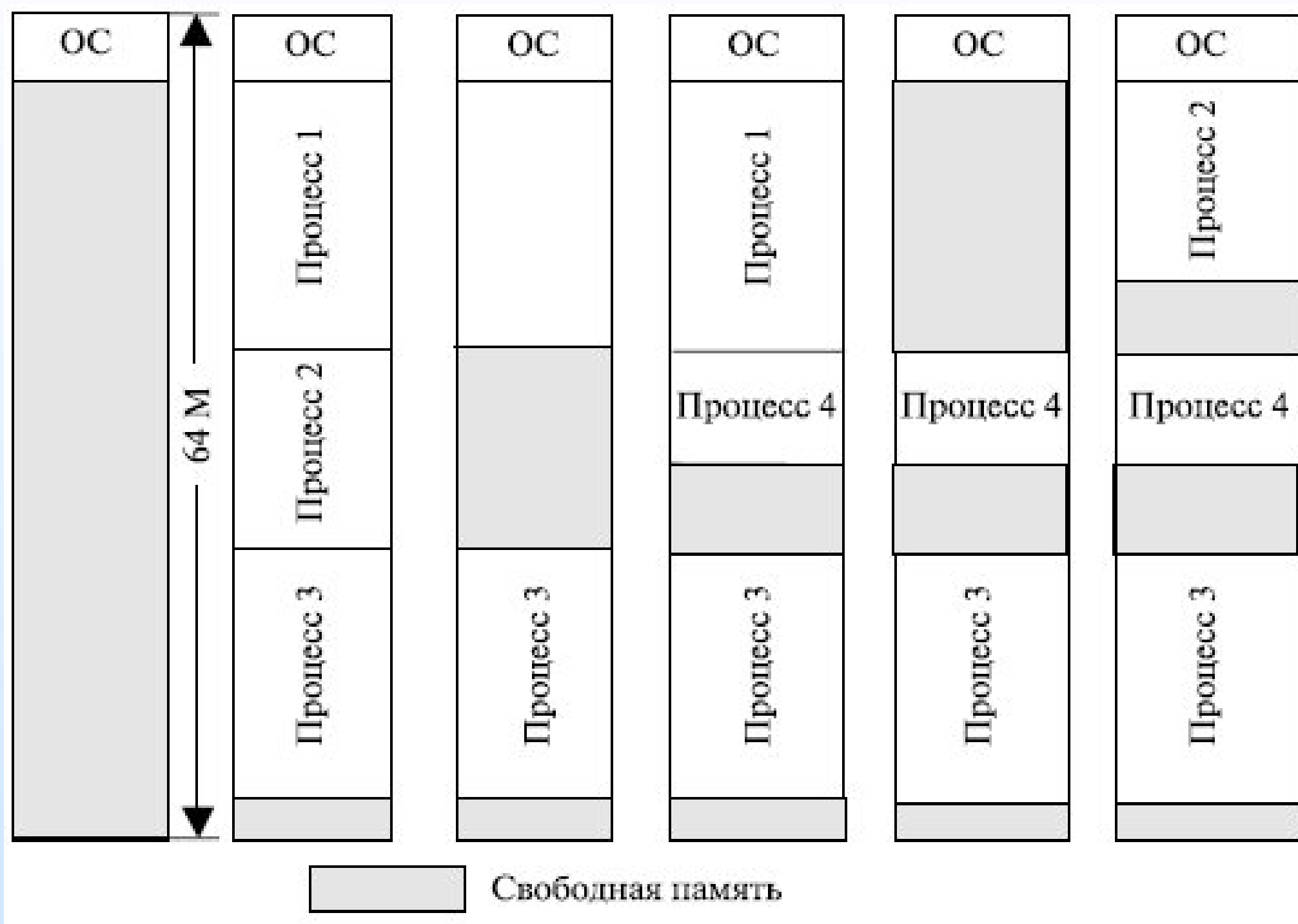
- Язык: python || C++ || Pascal
- Версия интерпретатора python: 2.7.12
- Компилятор C++: g++ 4.2.1 x86_64-apple-darwin16.4.0
- Консоль OS X



Где хранятся данные

- Жесткий диск
- ОЗУ
- Регистры

Программы в памяти





Разделение на сегменты

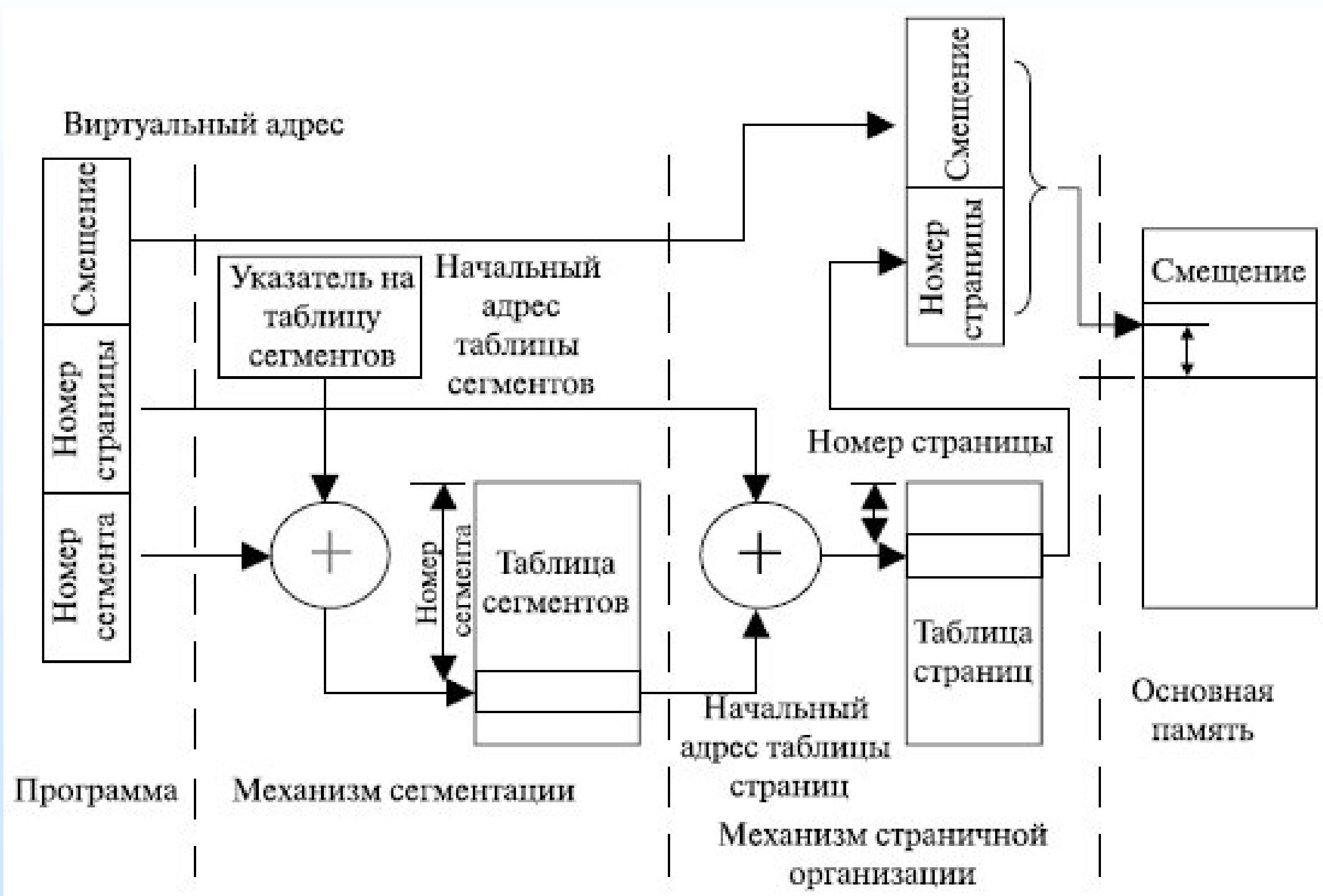
Виртуальное адресное пространство



Таблица кодировки символов достигла таблицы с исходным текстом

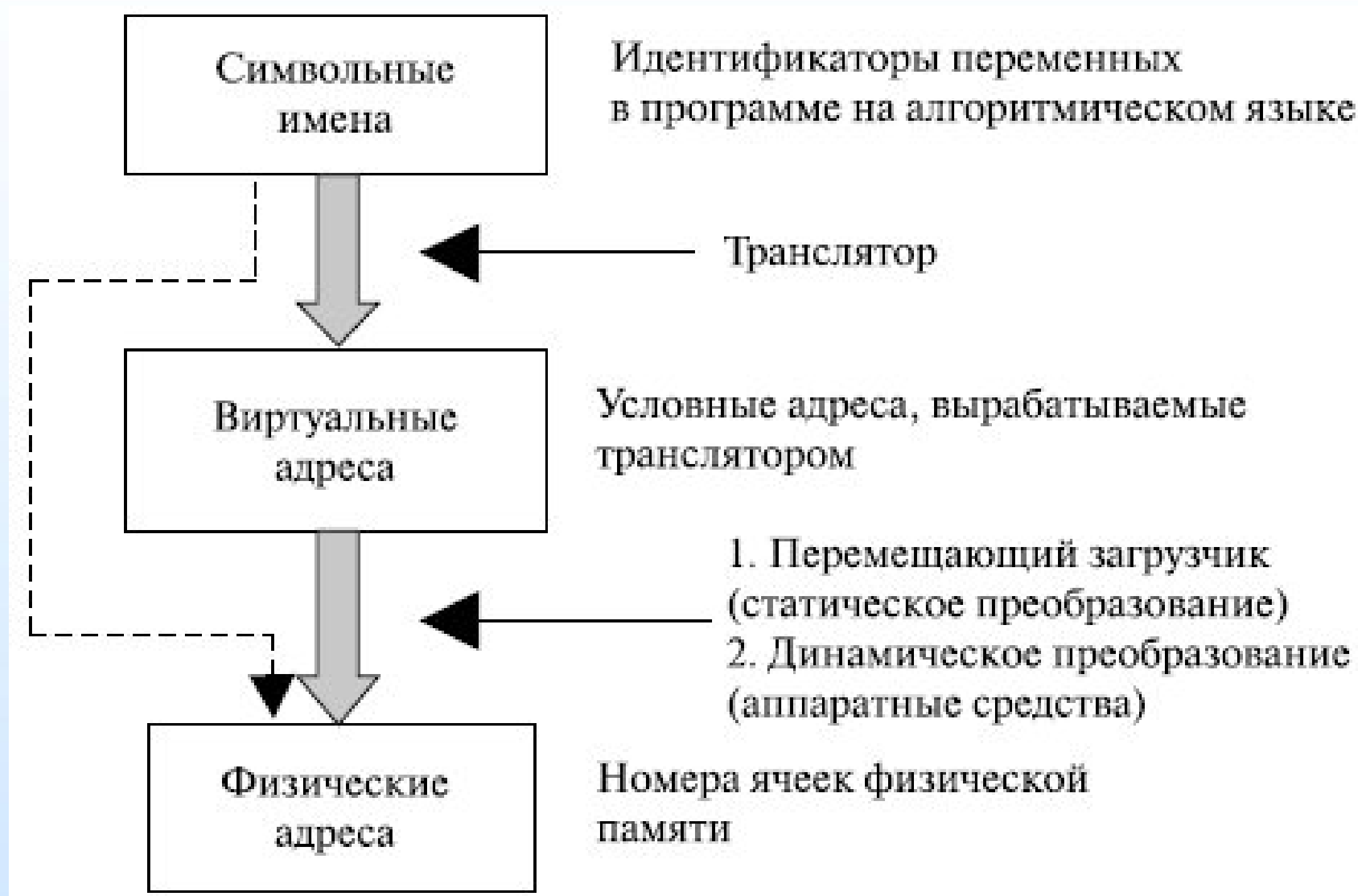


Сегментно-страничная организация памяти





Переменные



Данные

```
int x = 10;
```

Адрес	Данные
1000	10
1001	?
1002	?
1003	?
...	...
2000	x
2001	1000
2002	?
2003	?



Данные

```
int x = 10;  
int* px = &x;
```

Адрес	Данные
1000	10
1001	1000
1002	?
1003	?
...	...
2000	x
2001	1000
2002	px
2003	1001



Типы данных

- Целочисленные
 - Знаковые char, short, int
 - Беззнаковые uint8_t, uint16_t, uint32_t
- Вещественные float, double
- Указатели void*



Строки

```
char c = 'a';  
char* s = "It's a string";
```

Адрес	Данные
1000	'a'
1001	'l'
1002	't'
1003	'"
...	...
2000	c
2001	1000
2002	s
2003	1001



Структуры

```
struct point{  
    double x;  
    double y;  
};  
point a;  
a.x = 10;  
a.y = 15;
```

Адрес	Данные
1000	10
1001	15
1002	?
1003	?
...	...
2000	a
2001	1000
2002	?
2003	?



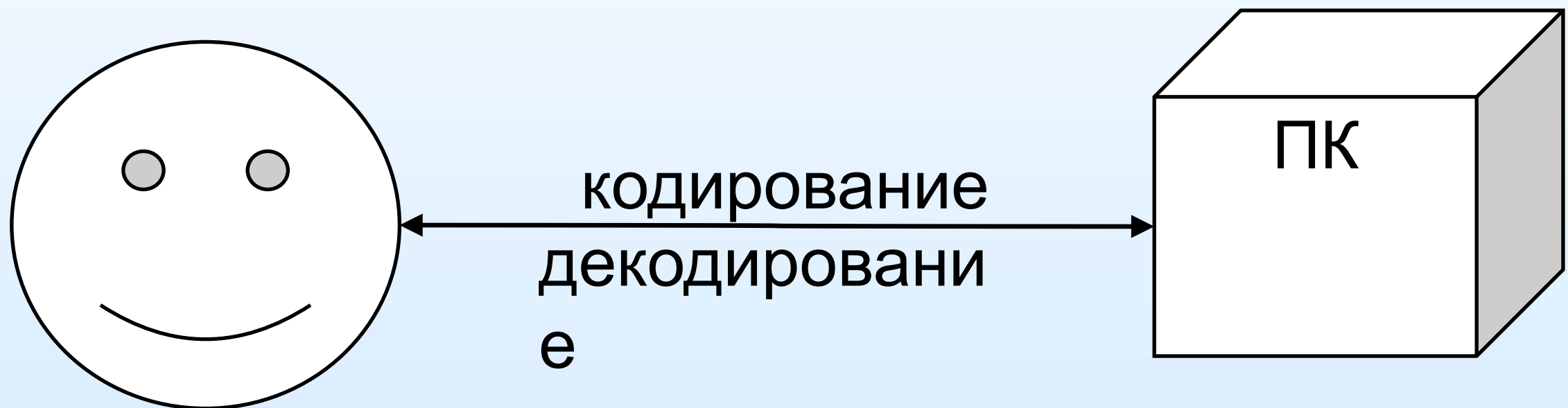
Указатели и функции

```
>>> def my_func1(x):  
...     return x+1  
...  
>>> def my_func2(x):  
...     return x-1  
...  
>>> def my_func3(y, x):  
...     return y(x)  
...  
>>> x = 10  
>>> my_func3(my_func1, x)  
11  
>>> my_func3(my_func2, x)  
9
```

Уровни абстракции

- Интуитивные структуры
- Абстрактные(логические) структуры
- Конкретные структуры

Кодирование





Задачи кодирования

- удобство физической реализации;
- удобство восприятия;
- высокая скорость передачи и обработки;
- экономичность, т.е. уменьшение избыточности сообщения;
- надежность, т.е. защита от случайных искажений;
- сохранность, т.е. защита от нежелательного доступа к информации.

Виды кодирования

- Код фиксированной длины
- Код переменной длины
 - Префиксный код
 - Постфиксный код



Пример кода фиксированной длины

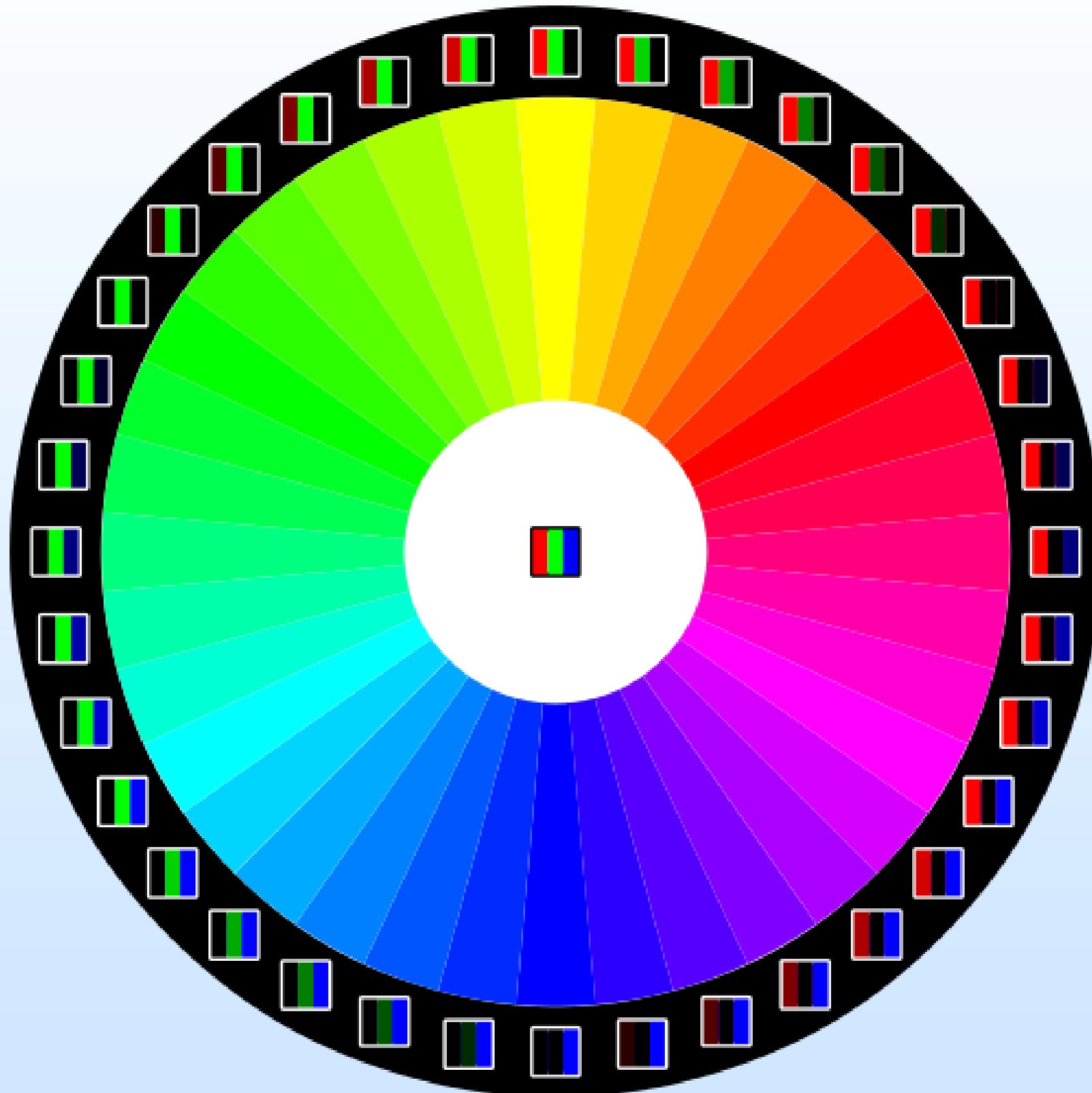
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	TAB	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL



Пример кода переменной длины

- $c(a)=00$
- $c(b)=01$
- $c(c)=1$
- $c*(abacaba)=0001001000100$
- 00 01 00 1 00 01 00

Кодирование изображений



JPEG && MP3



Спасибо за внимание!

aamaksutov@mephi.ru



ИНФОРМАТИКА

Структуры данных. Списки, массивы, таблицы, словари

Максутов Артем Артурович,
Ассистент кафедры 12

Москва, 2017

Сложности

- $O(n)$ — линейная сложность
- Такой сложностью обладает, например, алгоритм поиска наибольшего элемента в не отсортированном массиве. Нам придётся пройти по всем n элементам массива, чтобы понять, какой из них максимальный.

- $O(\log n)$ — логарифмическая сложность
- Простейший пример — бинарный поиск. Если массив отсортирован, мы можем проверить, есть ли в нём какое-то конкретное значение, методом деления пополам. Проверим средний элемент, если он больше искомого, то отбросим вторую половину массива — там его точно нет. Если же меньше, то наоборот — отбросим начальную половину. И так будем продолжать делить пополам, в итоге проверим $\log n$ элементов.



- $O(n^2)$ — квадратичная сложность
- Такую сложность имеет, например, алгоритм сортировки вставками. В канонической реализации он представляет из себя два вложенных цикла: один, чтобы проходить по всему массиву, а второй, чтобы находить место очередному элементу в уже отсортированной части. Таким образом, количество операций будет зависеть от размера массива как $n * n$, т. е. n^2 .

Примеры сложностей

размер сложность	10	20	30	40	50	60
n	0,00001 сек.	0,00002 сек.	0,00003 сек.	0,00004 сек.	0,00005 сек.	0,00005 сек.
n^2	0,0001 сек.	0,0004 сек.	0,0009 сек.	0,0016 сек.	0,0025 сек.	0,0036 сек.
n^3	0,001 сек.	0,008 сек.	0,027 сек.	0,064 сек.	0,125 сек.	0,216 сек.
n^5	0,1 сек.	3,2 сек.	24,3 сек.	1,7 минут	5,2 минут	13 минут
2^n	0,0001 сек.	1 сек.	17,9 минут	12,7 дней	35,7 веков	366 веков
3^n	0,059 сек.	58 минут	6,5 лет	3855 веков	2×10^8 веков	$1,3 \times 10^{13}$ веков

Массив – множество элементов, для которого упорядоченное множество целых чисел однозначно определяет позицию каждого элемента массива.

`int A[3] = {3,6,2};`

`std::vector<int> A = {3,6,2};`
`A = [3,6,2]`

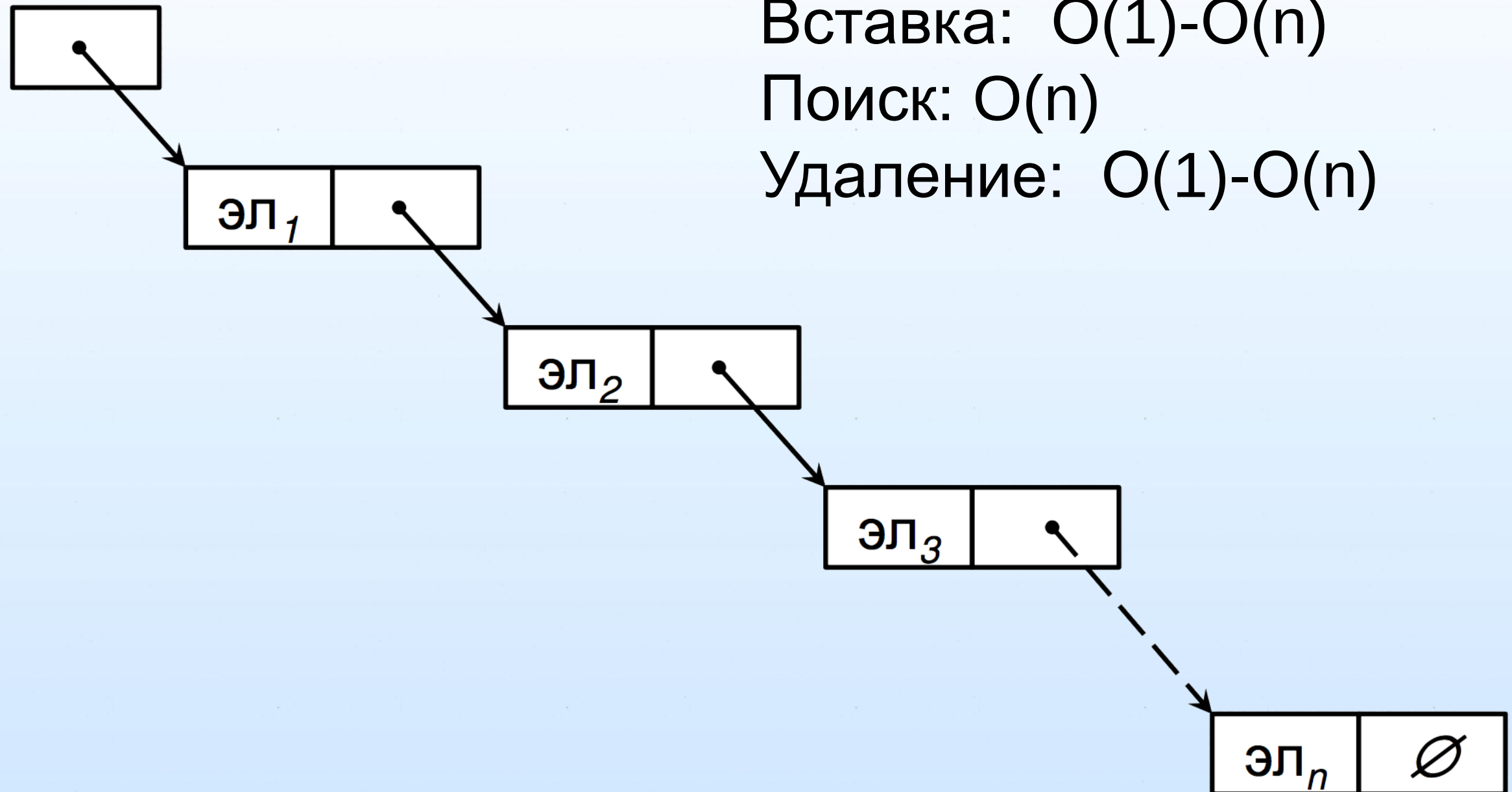
Индекс	Значение
0	3
1	6
2	2

Вставка: $O(1)$ - $O(n)$

Поиск: $O(\log(n))$ - $O(n)$

Удаление: $O(1)$ - $O(n)$

Список



Вставка: $O(1)$ - $O(n)$

Поиск: $O(n)$

Удаление: $O(1)$ - $O(n)$

(№ 31) В физической лаборатории проводится долговременный эксперимент по изучению гравитационного поля Земли. По каналу связи каждую минуту в лабораторию передаётся положительное целое число – текущее показание прибора «Сигма 2015». Количество передаваемых чисел в серии известно и не превышает 10 000. Все числа не превышают 1000. Временем, в течение которого происходит передача, можно пренебречь.

Необходимо вычислить «бета-значение» серии показаний прибора – минимальное чётное произведение двух показаний, между моментами передачи которых прошло не менее 6 минут. Если получить такое произведение не удаётся, ответ считается равным -1 .

Задача А. Напишите программу для решения поставленной задачи, в которой входные данные будут запоминаться в массиве, после чего будут проверены все возможные пары элементов. Максимальная оценка за выполнение задания А – 2 балла.

Задача Б. Напишите программу для решения поставленной задачи, которая будет эффективна как по времени, так и по памяти (или хотя бы по одной из этих характеристик).

Входные данные представлены следующим образом. В первой строке задаётся число N – общее количество показаний прибора. Гарантируется, что $N > 6$. В каждой из следующих N строк задаётся одно положительное целое число – очередное показание прибора.



12	45	5	3	17	23	21	20	19	18	17
----	----	---	---	----	----	----	----	----	----	----

rez=252 minodd=-1 mineven=12

12	45	5	3	17	23	21	20	19	18	17
----	----	---	---	----	----	----	----	----	----	----

rez=228 minodd=5 mineven=12

12	45	5	3	17	23	21	20	19	18	17
----	----	---	---	----	----	----	----	----	----	----

rez=54 minodd=3 mineven=12



```
1  N = int(input())
2  a=[]
3  for i in xrange(5):
4      a.append(int(input()))
5  mineven=-1
6  minodd=-1
7  m=-1
8  for i in xrange(N-5):
9      if a[i%5]%2:
10         if minodd>0:
11             if a[i%5] < minodd:
12                 minodd = a[i%5]
13         else:
14             minodd = a[i%5]
15     else:
16         if mineven>0:
17             if a[i%5] < mineven:
18                 mineven = a[i%5]
19         else:
20             mineven = a[i%5]
21     a[i%5] = int(input())
22     if m<0:
23         m = mineven * a[i%5]
24     if m < 0 and a[i%5] % 2 == 0:
25         m = minodd * a[i%5]
26     tmp = mineven*a[i%5]
27     if tmp < m and tmp > 0:
28         m = mineven*a[i%5]
29     tmp = minodd*a[i%5]
30     if tmp < m and tmp%2==0 and tmp > 0:
31         m = minodd*a[i%5]
32     if m>0 and m%2==0:
33         print m
34     else:
35         print -1
```




```
1  N = int(input())
2  a=[]
3  for i in xrange(5):
4      a.append(int(input()))
5  mineven=-1
6  minodd=-1
7  m=-1
```



```
8  for i in xrange(N-5):
9      if a[i%5]%2:
10         if minodd>0:
11             if a[i%5] < minodd:
12                 minodd = a[i%5]
13         else:
14             minodd = a[i%5]
15     else:
16         if mineven>0:
17             if a[i%5] < mineven:
18                 mineven = a[i%5]
19         else:
20             mineven = a[i%5]
21     a[i%5] = int(input())
```



```
22     if m<0:
23         m = mineven * a[i%5]
24     if m < 0 and a[i%5] % 2 == 0:
25         m = minodd * a[i%5]
26     tmp = mineven*a[i%5]
27     if tmp < m and tmp > 0:
28         m = mineven*a[i%5]
29     tmp = minodd*a[i%5]
30     if tmp < m and tmp%2==0 and tmp > 0:
31         m = minodd*a[i%5]
32 if m>0 and m%2==0:
33     print m
34 else:
35     print -1
```

Таблицы

Ключ	Информация
ProGamer	Вася
Dondo	Петя
...	
...	
...	

Таблицы

Ключ	Информация
ProGamer	Вася
Dondo	Тоже Вася
...	
Noob	Вася
...	

```
MacBook-Pro-Artem:~ darkness$ python3
```

```
>>> a = {"ProGamer":"Вася", "Dondo":"Тоже Вася"}
```

```
>>> a
```

```
{'Dondo': 'Тоже Вася', 'ProGamer': 'Вася'}
```



Словарь

```
>>> a['Noob'] = 'Вася'
```

```
>>> a
```

```
{'Dondo': 'Тоже Вася', 'Noob': 'Вася', 'ProGamer':  
'Вася'}
```

```
>>> a['Dondo'] = 'не Вася'
```

```
>>> a
```

```
{'Dondo': 'не Вася', 'Noob': 'Вася', 'ProGamer': 'Вася'}
```



(№ 4) В фрагменте базы данных представлены сведения о родственных отношениях. На основании приведённых данных определите, сколько прямых потомков (т.е. детей и внуков) Павленко А.К. упомянуты в таблице 1.

Таблица 1		
ID	Фамилия И.О.	Пол
2146	Кривич Л.П.	Ж
2155	Павленко А.К.	М
2431	Хитрук П.А.	М
2480	Кривич А.А.	М
2302	Павленко Е.А.	Ж
2500	Сокол Н.А.	Ж
3002	Павленко И.А.	М
2523	Павленко Т.Х.	Ж
2529	Хитрук А.П.	М
2570	Павленко П.И.	М
2586	Павленко Т.И.	Ж
2933	Симонян А.А.	Ж
2511	Сокол В.А.	Ж
3193	Биба С.А.	Ж
...	...	...

Таблица 2	
ID_Родителя	ID_Ребёнка
2146	2302
2146	3002
2155	2302
2155	3002
2302	2431
2302	2511
2302	3193
3002	2586
3002	2570
2523	2586
2523	2570
2529	2431
2529	2511
2529	3193
...	...

(№ 4) В фрагменте базы данных представлены сведения о родственных отношениях. На основании приведённых данных определите, сколько прямых потомков (т.е. детей и внуков) Павленко А.К. упомянуты в таблице 1.

Таблица 1		
ID	Фамилия И.О.	Пол
2146	Кривич Л.П.	Ж
2155	Павленко А.К.	М
2431	Хитрук П.А.	М
2480	Кривич А.А.	М
2302	Павленко Е.А.	Ж
2500	Сокол Н.А.	Ж
3002	Павленко И.А.	М
2523	Павленко Т.Х.	Ж
2529	Хитрук А.П.	М
2570	Павленко П.И.	М
2586	Павленко Т.И.	Ж
2933	Симонян А.А.	Ж
2511	Сокол В.А.	Ж
3193	Биба С.А.	Ж
...	...	...

Таблица 2	
ID_Родителя	ID_Ребёнка
2146	2302
2146	3002
2155	2302
2155	3002
2302	2431
2302	2511
2302	3193
3002	2586
3002	2570
2523	2586
2523	2570
2529	2431
2529	2511
2529	3193
...	...

(№ 4) В фрагменте базы данных представлены сведения о родственных отношениях. На основании приведённых данных определите, сколько прямых потомков (т.е. детей и внуков) Павленко А.К. упомянуты в таблице 1.

Таблица 1		
ID	Фамилия И.О.	Пол
2146	Кривич Л.П.	Ж
2155	Павленко А.К.	М
2431	Хитрук П.А.	М
2480	Кривич А.А.	М
2302	Павленко Е.А.	Ж
2500	Сокол Н.А.	Ж
3002	Павленко И.А.	М
2523	Павленко Т.Х.	Ж
2529	Хитрук А.П.	М
2570	Павленко П.И.	М
2586	Павленко Т.И.	Ж
2933	Симонян А.А.	Ж
2511	Сокол В.А.	Ж
3193	Биба С.А.	Ж
...	...	...

Таблица 2	
ID_Родителя	ID_Ребёнка
2146	2302
2146	3002
2155	2302
2155	3002
2302	2431
2302	2511
2302	3193
3002	2586
3002	2570
2523	2586
2523	2570
2529	2431
2529	2511
2529	3193
...	...

Спасибо за внимание!
Все вопросы по адресу:
AAMaksutov@mephi.ru



ИНФОРМАТИКА

Основы теории множеств

Максутов Артем Артурович,
Ассистент кафедры 12

Москва, 2017



Определение

- Теория множеств — раздел математики, в котором изучаются общие свойства множеств — совокупностей элементов произвольной природы, обладающих каким-либо общим свойством.

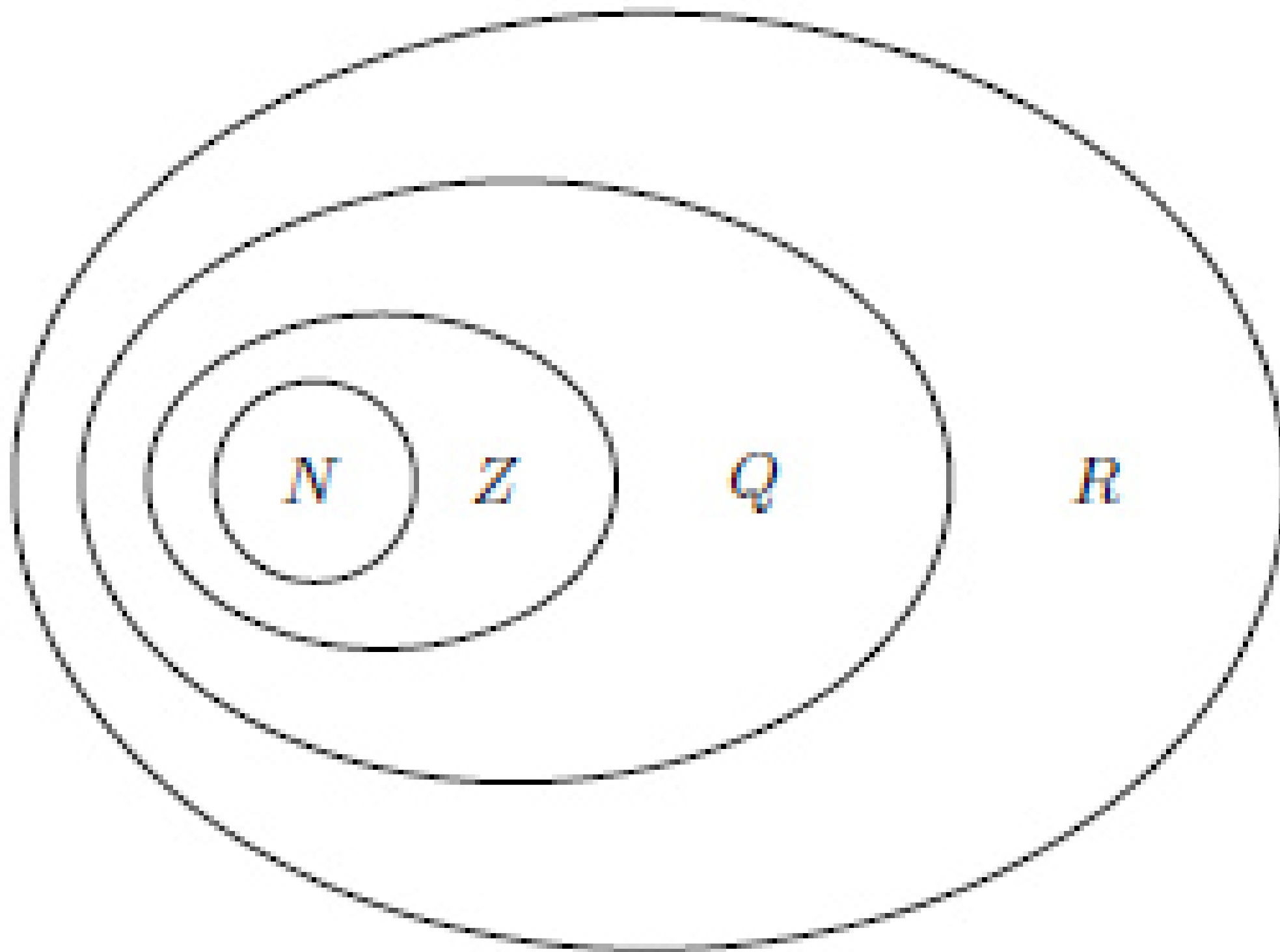


Отношение принадлежности

- Отношение принадлежности множества (обозначается как $x \in A$ — « x есть элемент множества A », « x принадлежит множеству A »)
- Пустое множество, обычно обозначается символом $\emptyset$ — множество, не содержащее ни одного элемента.
- Подмножество и надмножество — соотношения включения одного множества в другое (обозначаются соответственно $A \subseteq B$ и $A \supseteq B$ для нестрогого включения и $A \subset B$ и $A \supset B$ — для строгого).



Множества



Множества

$A=\{1,2,3,5,7\}$ — множество чисел

$X=\{x_1, x_2, \dots, x_n\}$ — множество некоторых
элементов $x_1, x_2, \dots, x_n$

$N=\{1, 2, \dots, n\}$ — множество натуральных чисел

$Z=\{0, \pm 1, \pm 2, \dots, \pm n\}$ — множество целых чисел



Способы задания множеств

1) Перечислением элементов

$$A = \{a, b, c, d\} .$$

2) Описанием характеристических свойств.

$$A = \{x | x < 10 \text{ и } x \in N\}.$$

Пример

$x < m$ $y > n$	$x \geq m \ \&\& \ x \leq n$ $y > n$	$x > m$ $y > n$
$x < m$ $y \geq m \ \&\& \ y \leq n$	$x \geq m \ \&\& \ x \leq n$ $y \geq m \ \&\& \ y \leq n$	$x > m$ $y \geq m \ \&\& \ y \leq n$
$x < m$ $y < m$	$x \geq m \ \&\& \ x \leq n$ $y < m$	$x > m$ $y < m$



Операции над множествами

- Два множества **A** и **B** равны ($A=B$), если они состоят из одних и тех же элементов.
- Например, если $A=\{1,2,3,4\}$, $B=\{3,1,4,2\}$ то $A=B$.



Операции над множествами

- **Объединением (суммой)** множеств A и B называется множество $A \cup B$, элементы которого принадлежат хотя бы одному из этих множеств.
- Например, если $A=\{1,2,4\}$, $B=\{3,4,5,6\}$, то $A \cup B = \{1,2,3,4,5,6\}$



Операции над множествами

- **Пересечением (произведением)** множеств A и B называется множество $A \cap B$, элементы которого принадлежат как множеству A , так и множеству B .
- Например, если $A=\{1,2,4\}$, $B=\{3,4,5,2\}$, то $A \cap B = \{2,4\}$



Операции над множествами

- **Разностью** множеств A и B называется множество $A \setminus B$, элементы которого принадлежат множеству A , но не принадлежат множеству B .
- Например, если $A = \{1, 2, 3, 4\}$, $B = \{3, 4, 5\}$, то $A \setminus B = \{1, 2\}$

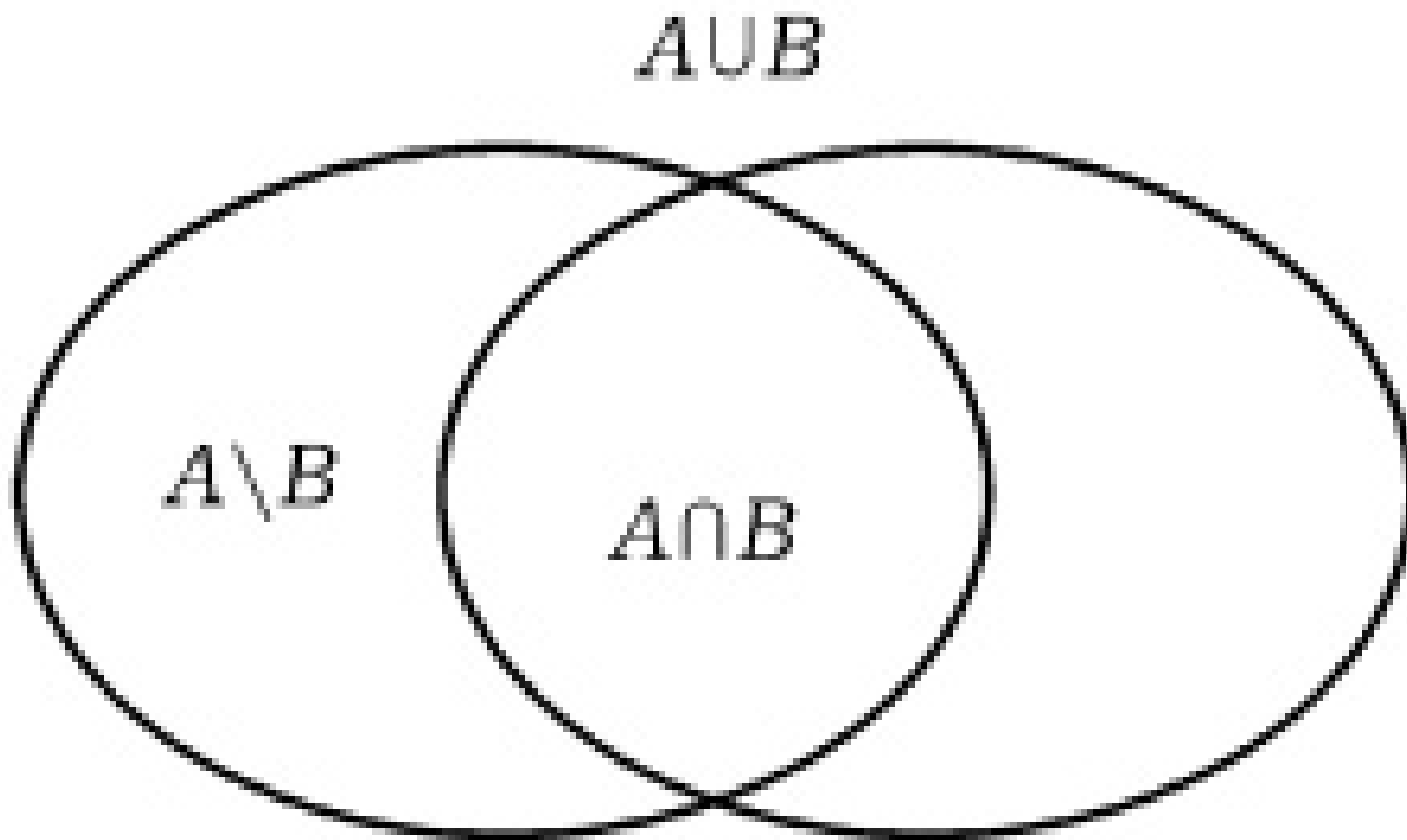


Операции над множествами

- **Симметричной разностью** множеств A и B называется множество $A \Delta B$, являющееся объединением разностей множеств $A \setminus B$ и $B \setminus A$, то есть $A \Delta B = (A \setminus B) \cup (B \setminus A)$.
- Например, если $A = \{1, 2, 3, 4\}$, $B = \{3, 4, 5, 6\}$, то $A \Delta B = \{1, 2\} \cup \{5, 6\} = \{1, 2, 5, 6\}$



Операции над множествами



Пример

По каналу связи передаётся последовательность положительных целых чисел, все числа не превышают 1000. Количество чисел известно, но может быть очень велико. Затем передаётся контрольное значение последовательности — наибольшее число R , удовлетворяющее следующим условиям:

R — произведение двух различных переданных элементов последовательности («различные» означает, что не рассматриваются квадраты переданных чисел, произведения различных элементов последовательности, равных по величине, допускаются);

R делится на 14.

Пример

На вход программе в первой строке подаётся количество чисел N . В каждой из последующих N строк записано одно натуральное число, не превышающее 1000. В последней строке записано контрольное значение.

Пример входных данных:

6
77
14
7
9
499
100
7700

Пример выходных данных для приведённого выше примера входных данных:

Вычисленное контрольное значение: 7700

Контроль пройден



Пример

- $CV = (7 * n) * (2 * m)$
- $CV = (14 * n) * m$

Пример

```

1  N = int(input())
2
3  m2 = 0
4  m7 = 0
5  m14 = 0
6  mx = 0
7  for i in xrange(N):
8      a = int(input())
9      if a % 2 == 0 and a % 7 == 0:
10         if m2 > m7:
11             if m7 < a:
12                 m7 = a
13         else:
14             if m2 < a:
15                 m2 = a
16         elif a % 2 == 0:
17             if a > m2:
18                 m2 = a
19         elif a % 7 == 0:
20             if a > m7:
21                 m7 = a
22
23         if a % 14 == 0 and a > m14:
24             m14 = a
25         elif a > mx:
26             mx = a
27
28     if int(input()) == max(m14*mx, m2*m7):
29         print "Yes"
30     else:
31         print "No"

```

```

1  Program HelloWorld(output);
2
3  var
4      i,N,m2,m7,m14,mx,a:integer;
5  begin
6      m2 := 0;
7      m7 := 0;
8      m14 := 0;
9      mx := 0;
10     readln(N);
11     for i:=1 to N do
12         begin
13             readln(a);
14             if ((a mod 2 = 0) and (a mod 7 = 0)) then begin
15                 if (m2>m7) then begin
16                     if (m7<a) then begin
17                         m7 := a;
18                     end
19                 end
20             else if (m2<a) then begin
21                 m2 := a;
22             end
23             end;
24             if a mod 2 = 0 then
25                 if a > m2 then
26                     m2 := a;
27             if a mod 7 = 0 then
28                 if a>m7 then
29                     m7 := a;
30             if ((a mod 14 = 0) and (a > m14)) then
31                 m14 := a
32             else
33                 if a > mx then
34                     mx := a;
35             end;
36             readln(a);
37             if (m2*m7 > m14*mx) then begin
38                 if (m2*m7 = a) then
39                     writeln('Yes')
40                 else
41                     writeln('No')
42                 end
43             else if (mx*m14 = a) then
44                 writeln('Yes')
45             else
46                 writeln('No');
47         end.

```


Пример

```
1  N = int(input())
2
3  m2 = 0
4  m7 = 0
5  m14 = 0
6  mx = 0
```

```
1  Program HelloWorld(output);
2
3  var
4      i, N, m2, m7, m14, mx, a: integer;
5  begin
6      m2 := 0;
7      m7 := 0;
8      m14 := 0;
9      mx := 0;
10     readln(N);
```



Пример

```
7  for i in xrange(N):
8      a = int(input())
9      if a % 2 == 0 and a % 7 == 0:
10         if m2 > m7:
11             if m7 < a:
12                 m7 = a
13         else:
14             if m2 < a:
15                 m2 = a
16     elif a % 2 == 0:
17         if a > m2:
18             m2 = a
19     elif a % 7 == 0:
20         if a > m7:
21             m7 = a
```

```
12  begin
13      readln(a);
14      if ((a mod 2 = 0) and (a mod 7 = 0)) then begin
15          if (m2 > m7) then begin
16              if (m7 < a) then begin
17                  m7 := a;
18              end
19          end
20      else if (m2 < a) then begin
21          m2 := a;
22      end
23      end;
24      if a mod 2 = 0 then
25          if a > m2 then
26              m2 := a;
27      if a mod 7 = 0 then
28          if a > m7 then
29              m7 := a;
```

Пример

```
22  
23     if a % 14 == 0 and a > m14:  
24         m14 = a  
25     elif a > mx:  
26         mx = a  
27
```

```
30 ▾     if ((a mod 14 = 0) and (a > m14)) then  
31         m14 := a  
32 ▾     else  
33 ▾         if a > mx then  
34             mx := a;
```



Пример

```
28  if int(input())==max(m14*mx, m2*m7):  
29      print "Yes"  
30  else:  
31      print "No"
```

```
36      readln(a);  
37  if (m2*m7 > m14*mx) then begin  
38      if (m2*m7 = a) then  
39          writeln('Yes')  
40      else  
41          writeln('No')  
42      end  
43  else if (mx*m14 = a) then  
44      writeln('Yes')  
45  else  
46      writeln('No');  
47  end.
```


Спасибо за внимание!
Все вопросы по адресу:
AAMaksutov@mephi.ru



ТЕМА

Алгоритмы: построение и визуализация

Максутов Артем Артурович,
Ассистент кафедры №12 «Компьютерные системы и технологии»

Москва, 2017



- $CV = m2 * m7$
- $CV = m14 * mx$
- $CV = m14_1 * m14_2$



end.
48) Определите, количество чисел K, для которых следующая программа выведет такой же результат, что и для K = 25:
var i, k: integer;
function F(x:integer):integer;
begin
 F:=x*x+8*x;
end;
begin
 i := 0;
 readln(K);
 while F(i) < K do
 i:=i+1;
 writeln(i);
end.

$$x^2 + 8x < 25$$

$$x < 3$$

$$F(3) = 33$$

$$F(2) = 20$$

$$K = (20, 33]$$



Алгоритм –
это точное и понятное предписание
(указание) исполнителю совершить
определенную последовательность
действий, направленных на решение
поставленной задачи.



Исполнитель выполняет алгоритм формально



Системой команд Исполнителя называется совокупность всех команд, которые может выполнить Исполнитель.

Совокупность всех действий, которые он может выполнить в ответ на эти команды, называется системой допустимых действий Исполнителя.

Среда – это обстановка, в которой работает исполнитель

Элементарное действие – действие, совершаемое исполнителем после вызова команды.

Отказы. Возникают при вызове команды в недопустимом для данной команды состоянии среды.



1. Дискретность (любой алгоритм можно разбить на отдельные простые команды – шаги)
2. Понятность (алгоритм должен быть понятен исполнителю)
3. Однозначность (команда алгоритма должна истолковываться однозначно)



4. Массовость (алгоритм можно применить для решения множества однотипных задач)

5. Результативность (алгоритм должен быть конечным и иметь результат)



СПОСОБЫ ОПИСАНИЯ АЛГОРИТМОВ





1. Словесный – описание алгоритма словами

- Рецепт приготовления блюда
- Алгоритм решения задачи
- ...



2. Графический – описание алгоритма рисунками, чертежами, схемами ...





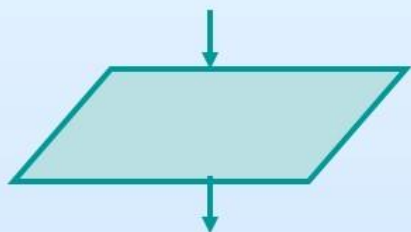
3. Блок - схема – описание алгоритма с помощью стандартных обозначений

Обозначение

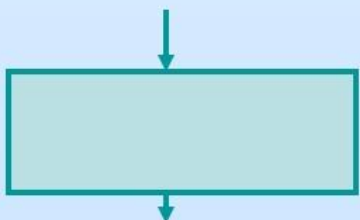


Название блоков

начало/конец



ввод/вывод данных

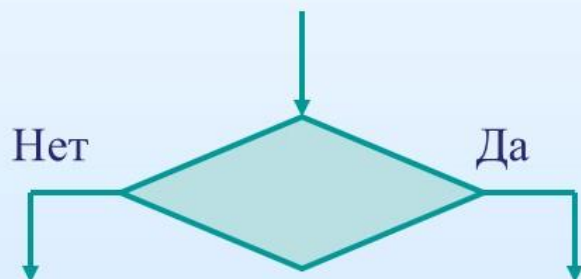


блок действия



3. Блок - схема – описание алгоритма с помощью стандартных обозначений

Обозначение



Название блоков

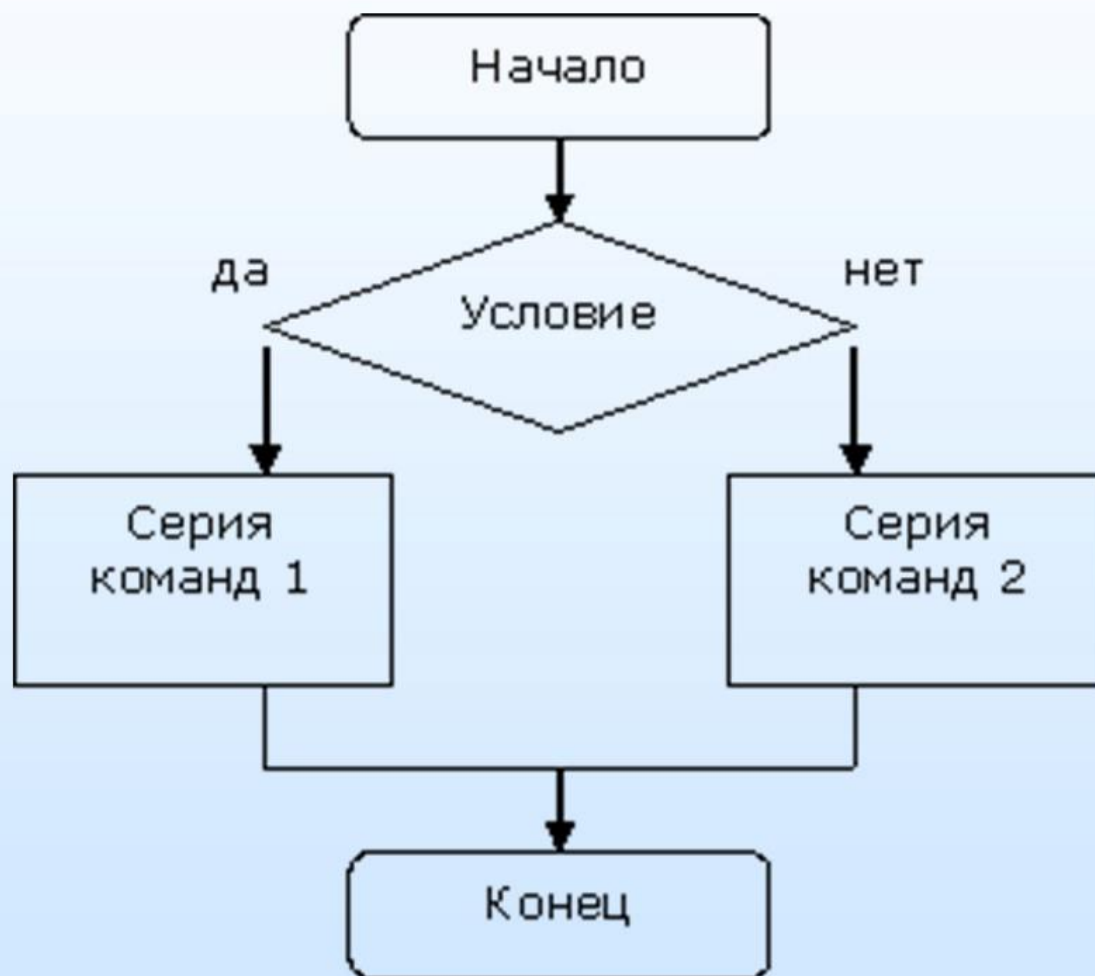
условие

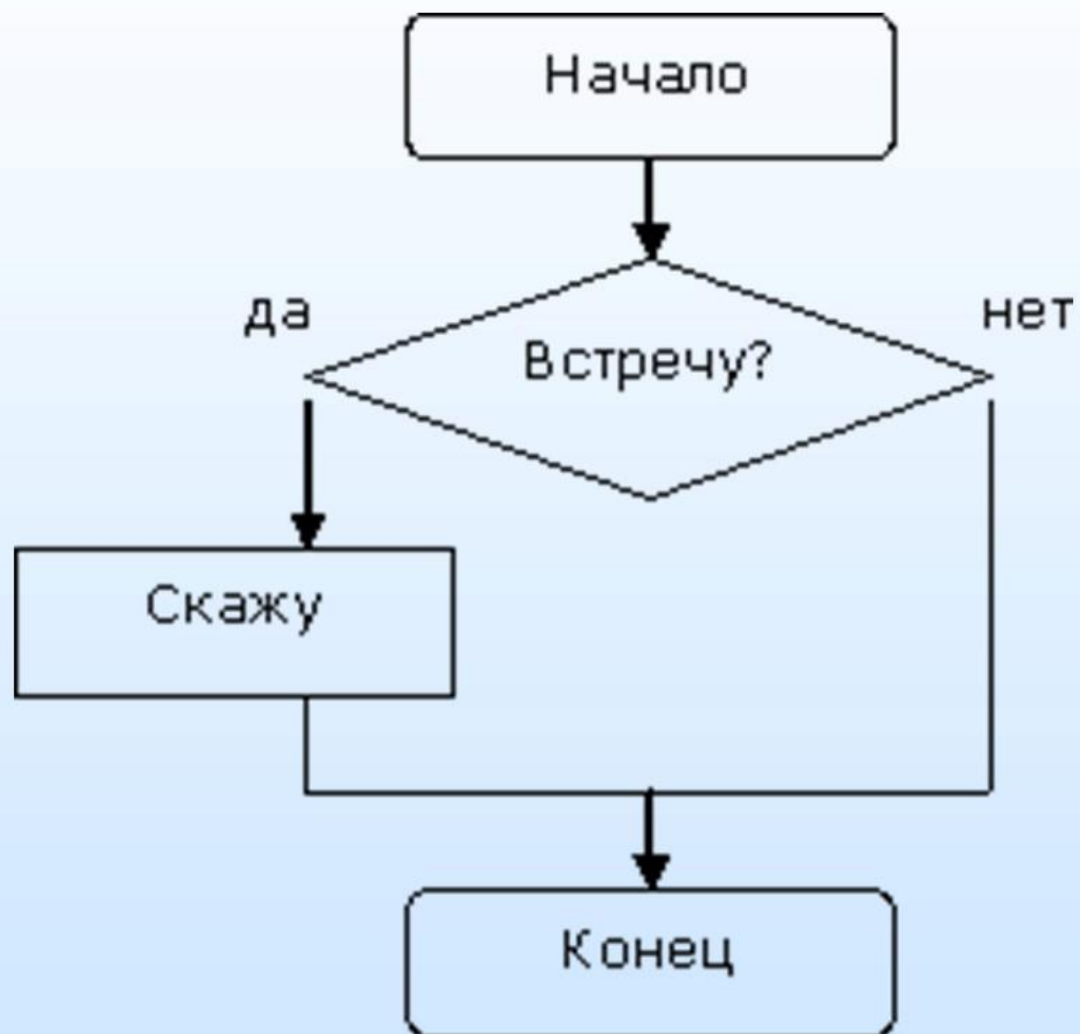


Типы алгоритмов

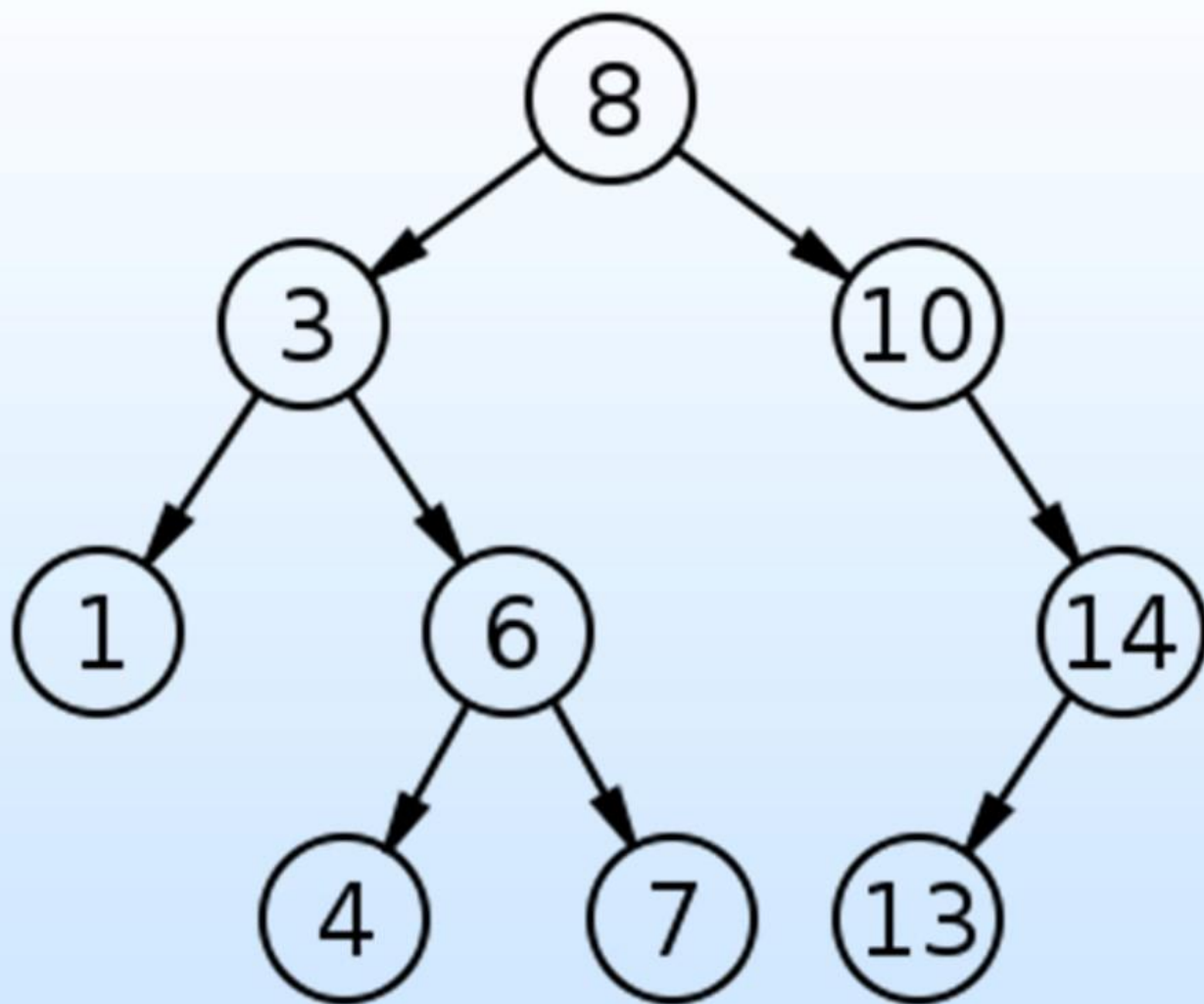
- Линейный алгоритм
- Алгоритм с ветвлением
- Циклический алгоритм
- Рекурсивный алгоритм















Исходная программа

Входные данные



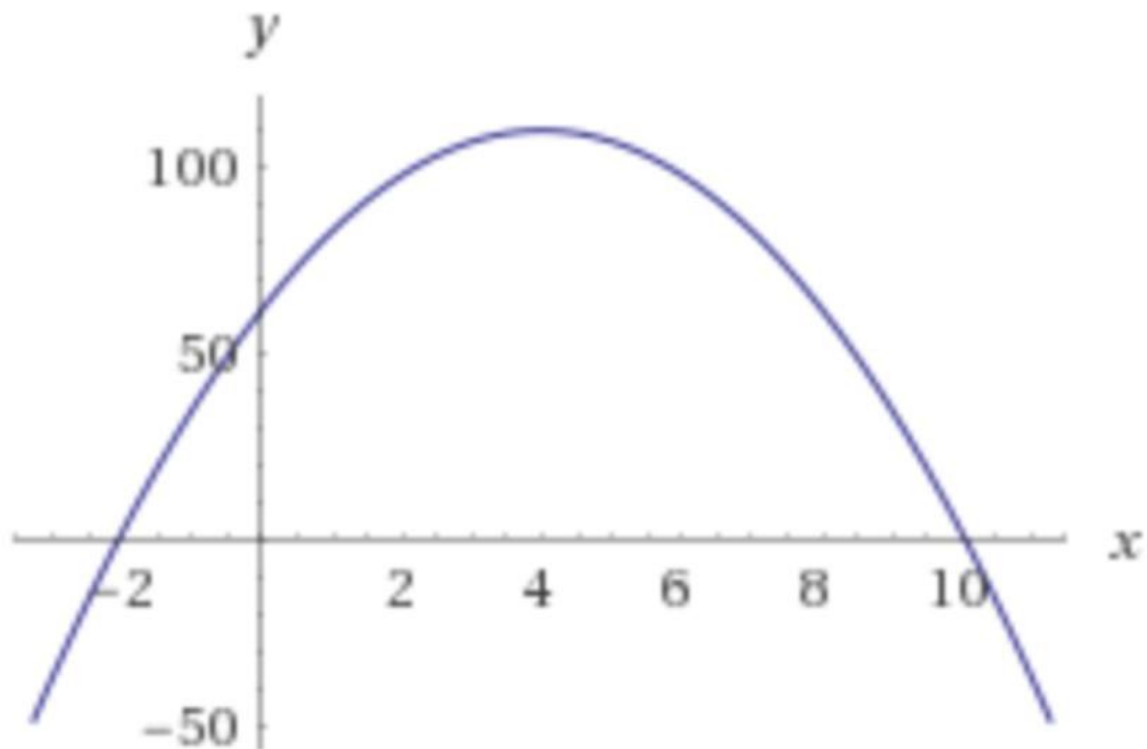
Результат



РОСАТОМ

19) Определите, какое число будет напечатано в результате выполнения следующего алгоритма:

```
var a,b,t,M,R:integer;  
Function F(x: integer):integer;  
begin  
  F := -3*(x-10)*(x+2)+2;  
end;  
BEGIN  
  a := -20; b := 20;  
  M := a; R:= F(a);  
  for t := a to b do  
    if (F(t)>R) then begin  
      M := t;  
      R:= F(t);  
    end;  
  write(M);  
END.
```





Спасибо за внимание!
Все вопросы по адресу:
AAMaksutov@mephi.ru

ИНФОРМАТИКА

Алгоритмы: джентельменский набор разработчика

Максутов Артем Артурович,
Ассистент кафедры 12

Москва, 2017

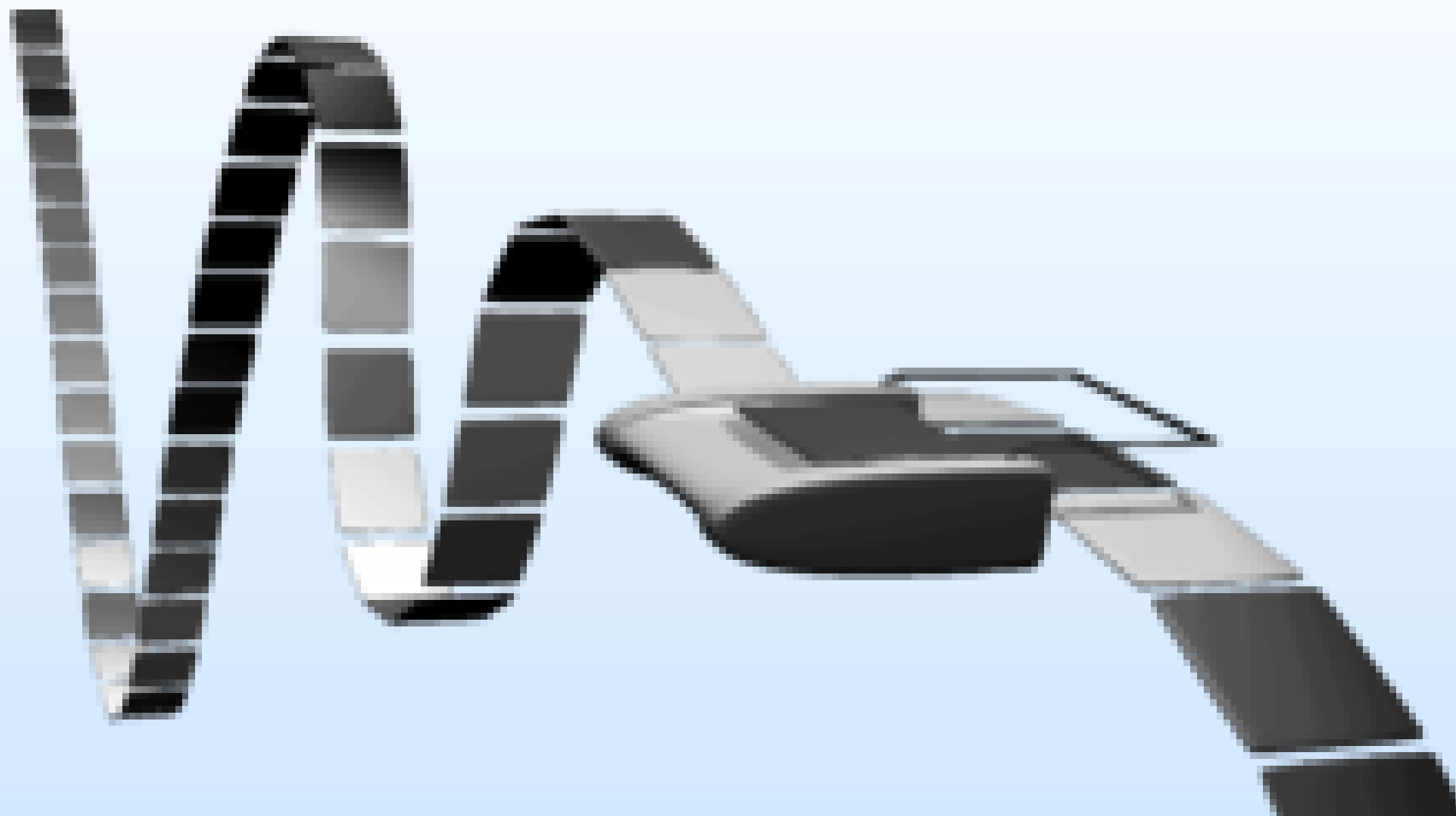


Классификация

- Алгоритмы поиска
- Алгоритмы сортировки
- Векторные операции
- Матрично-векторные операции
- Матричные операции
- Разложения матриц
- Решение систем линейных уравнений
- Решения спектральных задач
- Преобразование Фурье
- Алгебра многочленов
- Численные методы интегрирования
- Алгоритмы на графах
- Другие алгоритмы



Машина Тьюринга





Решение задачи

1. Содержательная постановка задачи.
2. Системный анализ.
3. Системный синтез (математическая постановка задачи)
4. Разработка или выбор программного обеспечения.
5. Решение задачи.



Алгоритмы поиска

Последовательный поиск

```
for i in xrange(N):  
    if a[i] == b:  
        do_something()
```

Бинарный поиск

```
def find(amin, amax, b):  
    med = amin + (amax-amin)/2  
    if a[med] == b:  
        dosomething()  
    if amax==amin and a[amin]!=b:  
        return  
    if a[med] > b:  
        find(amin, med, b)  
    else:  
        find(med, amax, b)
```



Алгоритмы поиска

```
/// Binary search function using Pascal
var
  a: array of integer = (1, 2, 3, 4, 5);

// Defining function
function binarySearch(arr: array of integer; toFind: integer): integer;
var
  first, last, mid: integer;
begin
  // Checking our array for emptiness.
  if ((arr = nil) or (length(arr) = 0)) then result := -1 else
  begin
    first := 0;
    last := length(arr) - 1;
    // Running binary search
    while (first < last) do
    begin
      mid := first + ((last - first) div 2);
      if (arr[mid] >= toFind) then last := mid else first := mid + 1;
    end;
    // After the end of the loop, lastIndex can point to the search item.
    // Otherwise the element is missing in the array.
    if (arr[last] = toFind) then result := last else result := -1;
  end;
end;

begin
  // Let's try out our function:
  writeln(binarySearch(a, 3)); // Output is: 2
  writeln(binarySearch(a, 1)); // Output is: 0
  writeln(binarySearch(a, 2)); // Output is: 1
  writeln(binarySearch(a, 5)); // Output is: 4
  writeln(binarySearch(a, 9)); // Output is: -1
  writeln(binarySearch(a, 0)); // Output is: -1
end.
```



Алгоритмы сортировки

ЦИКЛ ДЛЯ $J=1$ ДО $N-1$ ШАГ 1

$F=0$

ЦИКЛ ДЛЯ $I=1$ ДО $N-J$ ШАГ 1

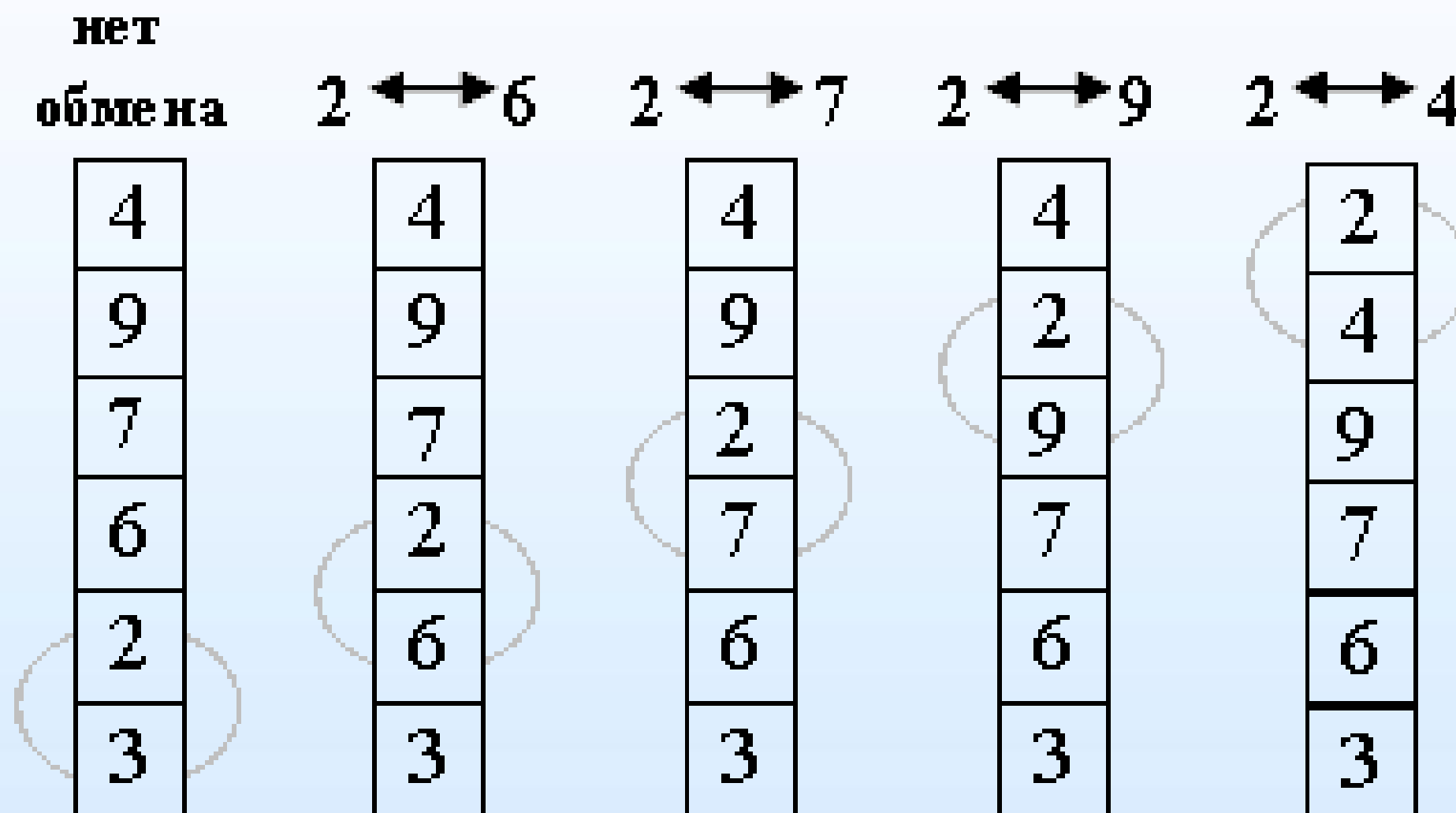
ЕСЛИ $A[I] > A[I+1]$ ТО ОБМЕН $A[I], A[I+1] : F=1$

СЛЕДУЮЩЕЕ I

ЕСЛИ $F=0$ ТО ВЫХОД ИЗ ЦИКЛА

СЛЕДУЮЩЕЕ J

Алгоритмы сортировки



Нулевой проход, сравниваемые пары выделены

3 1 5 8 1 0 4 6 6 7

3 1 5 8 0 1 4 6 6 7

3 1 5 0 8 1 4 6 6 7

3 1 0 5 8 1 4 6 6 7

3 0 1 5 8 1 4 6 6 7

0 3 1 5 8 1 4 6 6 7 Left=1

0 1 3 5 8 1 4 6 6 7

0 1 3 5 1 8 4 6 6 7

0 1 3 5 1 4 8 6 6 7

0 1 3 5 1 4 6 8 6 7

0 1 3 5 1 4 6 6 8 7

0 1 3 5 1 4 6 6 7 8 Right=10

0 1 3 1 5 4 6 6 7 8

0 1 1 3 5 4 6 6 7 8 Left=3

0 1 1 3 4 5 6 6 7 8



Пример

19

В программе используется одномерный целочисленный массив A с индексами от 0 до 9. Значения элементов равны 1, 2, 5, 8, 9, 3, 4, 0, 7, 6 соответственно, т.е. $A[0] = 1$, $A[1] = 2$ и т.д.

Определите значение переменной j после выполнения следующего фрагмента программы (записанного ниже на пяти языках программирования).

Бейсик	Python
<pre>j = 5 WHILE A(j) < A(j-1) t = A(j) A(j) = A(j-1) A(j-1) = t j = j - 1 WEND</pre>	<pre>j = 5 while A[j] < A[j-1]: A[j], A[j-1] = A[j-1], A[j] j -= 1</pre>
Алгоритмический язык	Паскаль
<pre>j := 5 нц пока A[j] < A[j-1] t := A[j] A[j] := A[j-1] A[j-1] := t j := j - 1 кц</pre>	<pre>j := 5; while A[j] < A[j-1] do begin t := A[j]; A[j] := A[j-1]; A[j-1] := t; j := j - 1; end;</pre>



Пример

20

Ниже на пяти языках программирования записан алгоритм. Получив на вход натуральное число x , этот алгоритм печатает число R . Укажите такое число x , при вводе которого алгоритм печатает двузначное число, сумма цифр которого равна 16. Если таких чисел x несколько, укажите наименьшее из них.

Бейсик	Python
<pre>DIM X,D,R AS LONG INPUT X R = 0 WHILE X>0 D = X MOD 10 R = 10*R + D X = X \ 10 WEND PRINT R</pre>	<pre>x = int(input()) R = 0 while x>0: d = x % 10 R = 10*R + d x = x // 10 print(R)</pre>
Алгоритмический язык	Паскаль
<pre><u>алг</u> <u>нач</u> <u>цел</u> x, d, R <u>ввод</u> x R := 0 <u>нц пока</u> x>0 d := mod(x, 10) R := 10*R + d x := div(x, 10) <u>кц</u> <u>вывод</u> R <u>кон</u></pre>	<pre>var x,d,R: longint; begin readln(x); R := 0; while x>0 do begin d := x mod 10; R := 10*R + d; x := x div 10 end; writeln(R) end.</pre>

Пример

Задание Б. Имеется набор данных, состоящий из пар положительных целых чисел. Необходимо выбрать из каждой пары ровно одно число так, чтобы сумма всех выбранных чисел не делилась на 3 и при этом была максимально возможной. Если получить требуемую сумму невозможно, в качестве ответа нужно выдать 0.

Напишите программу для решения этой задачи.

Постарайтесь сделать программу эффективной по времени и используемой памяти (или хотя бы по одной из этих характеристик).



Пример

Для варианта Б на вход программе в первой строке подаётся количество пар N ($1 \leq N \leq 100\,000$). Каждая из следующих N строк содержит два натуральных числа, не превышающих 10 000.

Пример входных данных для варианта Б:

```
6
1 3
5 12
6 9
5 4
3 3
1 1
```

Пример выходных данных для приведённых выше примеров входных данных:

```
32
```



Пример

$$result = sum - delta ,$$

где

$$delta = \begin{cases} 0, \text{ при } sum \bmod 3 \neq 0 \\ \min (\{d \mid d = |a1 - a2| : |a1 - a2| \bmod 3 \neq 0\}) \end{cases}$$



Пример

```
1  N = int(input())
2
3  sum = 0
4  delta = 0
5  for i in xrange(N):
6      a1, a2 = [int(x) for x in raw_input().split()]
7      sum += max(a1, a2)
8      if delta % 3 == 0:
9          delta = abs(a1-a2)
10     elif abs(a1-a2) % 3 > 0:
11         delta = min(delta, abs(a1-a2))
12
13  print sum if sum % 3 != 0 else sum - delta \
14      if sum - delta % 3 != 0 else 0
```


Спасибо за внимание!
Все вопросы по адресу:
AAMaksutov@mephi.ru



ИНФОРМАТИКА

Подходы к решению задач. Полный перебор, рекурсия, динамическое программирование

Максутов Артем Артурович,
Ассистент кафедры 12

Москва, 2017



Полный перебор

```
for i in range(n):  
    for j in range(m):  
        for k in range(l):  
            do_smth()
```



Полный перебор

do_smth()	do_smth()
-----------	-----------

do_smth()

for k in range(l):	for k in range(l):	for k in range(l):
--------------------	--------------------	--------------------

for j in range(n):	for j in range(n):	for j in range(n):
--------------------	--------------------	--------------------

for i in range(m):	for i in range(m):	for i in range(m):
--------------------	--------------------	--------------------

Рекурсия

рекурсия



Все

Картинки

Видео

Новости

Книги

Ещё

Настройки

Инструменты

Результатов: примерно 331 000 (0,44 сек.)

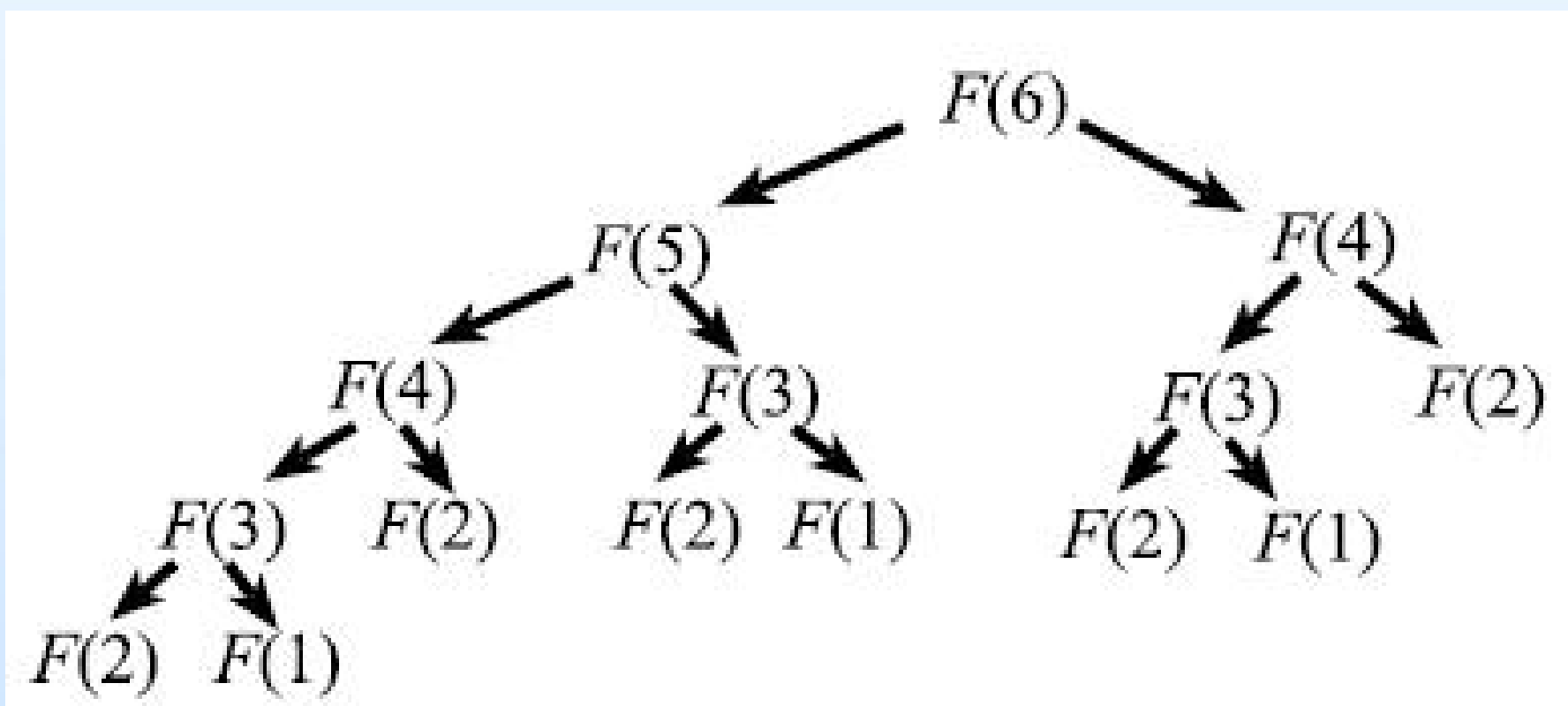
Возможно, вы имели в виду: **рекурсия**



Рекурсия

1, 1, 2, 3, 5, 8, 13...

$F(1) = 1$, $F(2) = 1$, $F(n) = F(n-2) + F(n-1)$, при $n > 2$

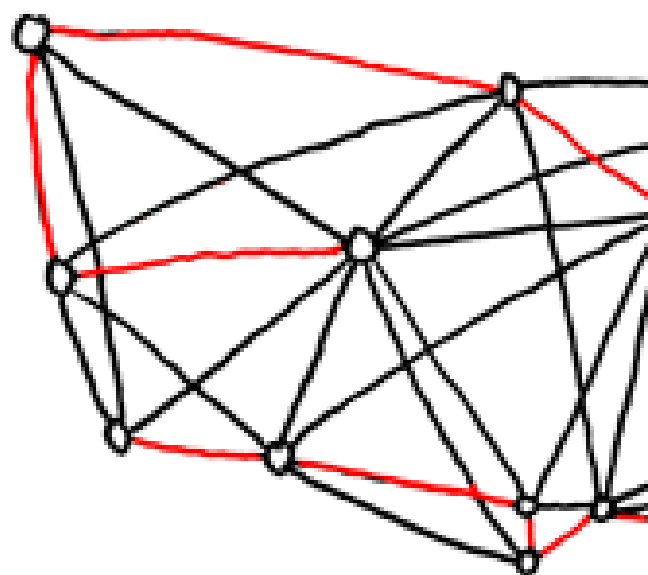




Динамическое программирование

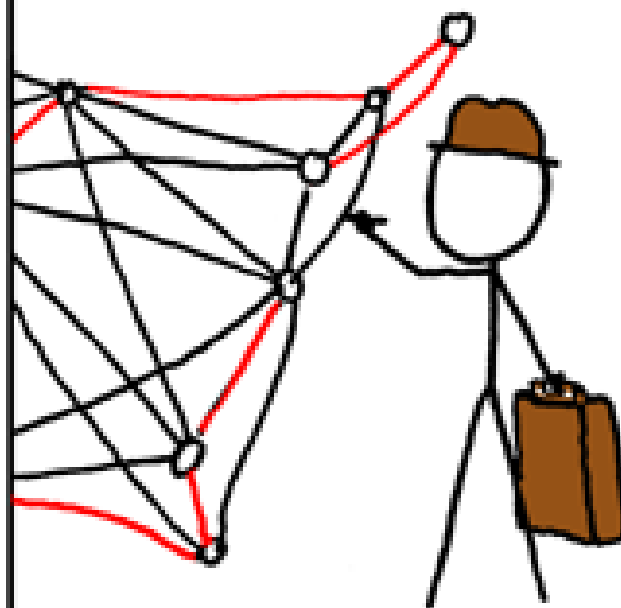
РЕШЕНИЕ
ПЕРЕБОРОМ:

$$O(n!)$$



АЛГОРИТМЫ
ДИНАМИЧЕСКОГО
ПРОГРАММИРОВАНИЯ:

$$O(n^2 2^n)$$



ПРОДАЖА ЧЕРЕЗ ЕВАУ:

$$O(1)$$

ВСЁ ЕЩЁ
ВЫЧИСЛЯЕШЬ ПУТЬ?

ЗАТКНИСЬ.

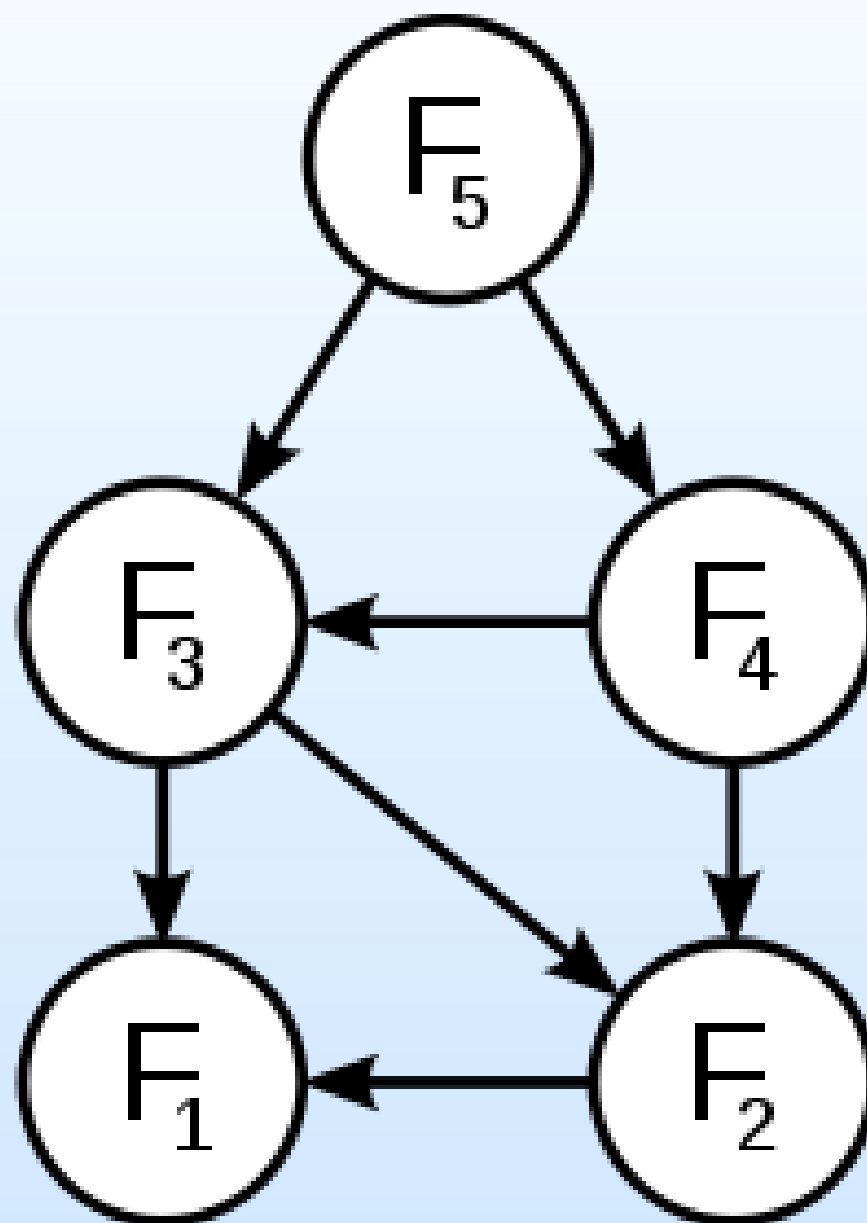




Динамическое программирование

Динамическое программирование
— способ решения сложных задач
путём разбиения их на более
простые подзадачи.

Граф решения задачи





Свойства задач

- перекрывающиеся подзадачи;
- оптимальная подструктура;
- ВОЗМОЖНОСТЬ запоминания решения часто встречающихся подзадач.

Пример

Два игрока, Паша и Валя, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Паша. За один ход игрок может добавить в кучу **один** камень или увеличить количество камней в куче **в два раза**. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16 или 30 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится не менее 20. Если при этом в куче оказалось не более 30 камней, то победителем считается игрок, сделавший последний ход. В противном случае победителем становится его противник. Например, если в куче было 17 камней и Паша удвоит количество камней в куче, то игра закончится, и победителем будет Валя. В начальный момент в куче было S камней, $1 \leq S \leq 19$.

Пример

1. а) При каких значениях числа S Паша может выиграть в один ход?
Укажите все такие значения и соответствующие ходы Паши.
б) У кого из игроков есть выигрышная стратегия при $S = 18, 17, 16$?
Опишите выигрышные стратегии для этих случаев.
2. У кого из игроков есть выигрышная стратегия при $S = 9, 8$? Опишите соответствующие выигрышные стратегии.
3. У кого из игроков есть выигрышная стратегия при $S = 7$? Постройте дерево всех партий, возможных при этой выигрышной стратегии (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход; в узлах — количество камней в позиции.



Пример

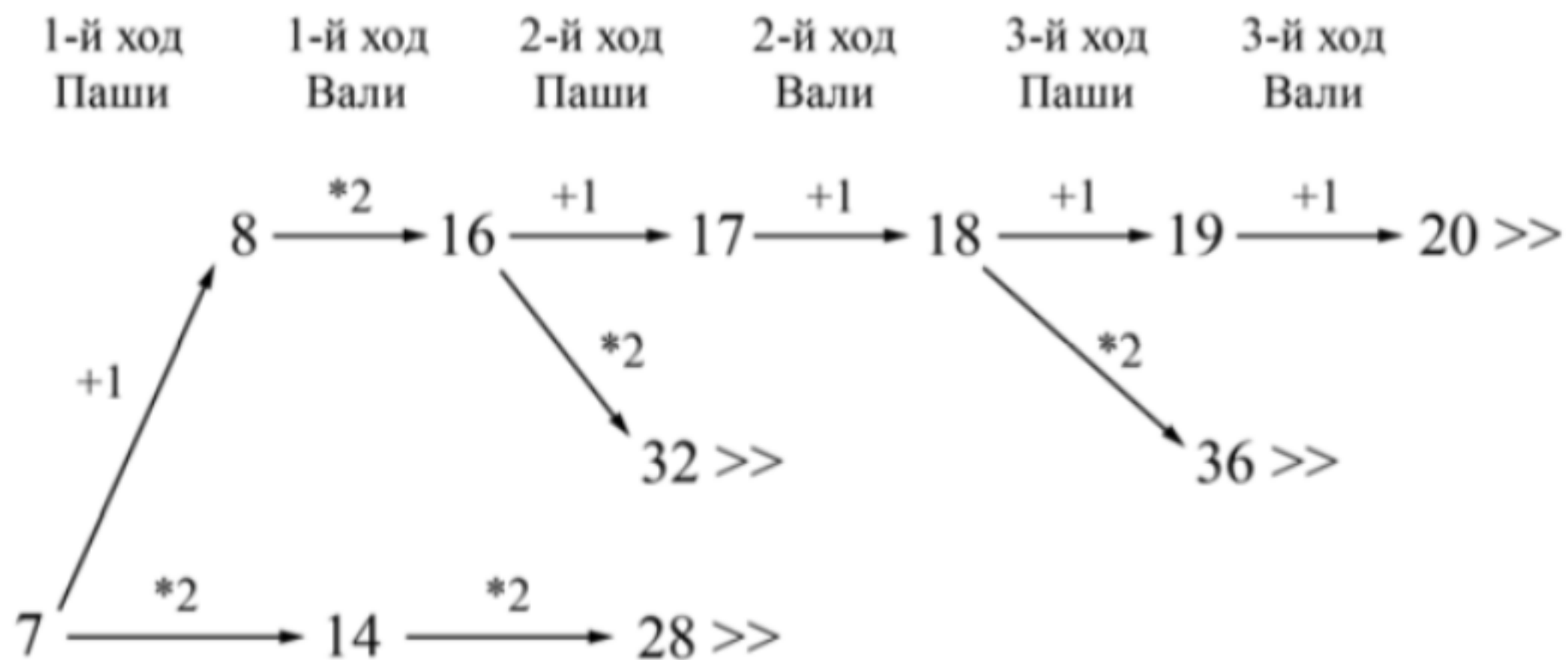
$$S' = \{W' \mid w' = w - 1 : w \in W\} \cup \{W'' \mid w'' = w/2 : w \in W\}$$

Таким образом множество S' состоит из элементов $\{[10,15], [19,29]\}$.

Однако, надо понимать, что во множестве S не могут находиться элементы множества W , т.к. это подразумевает выигрыш Вали на момент хода Паши.

Таким образом множество S имеет следующий вид:

$$S = S' - W = \{[10,15], 19\}$$



Пример

Имеется набор данных, состоящий из n пар положительных целых чисел. Необходимо выбрать из каждой пары ровно одно число так, чтобы сумма всех выбранных чисел не делилась на 3 и при этом была максимально возможной. Если получить требуемую сумму невозможно, в качестве ответа нужно выдать 0.

Пример

Входные данные:

На вход программе в первой строке подаётся количество пар N ($1 \leq N \leq 100\,000$). Каждая из следующих N строк натуральное число, не превышающее 10 000.

Пример входных данных для варианта Б: 6

13

5 12

69

54

33

11

Пример выходных данных для приведённых выше примеров
входных
данных: 32



Пример

$$sum = \sum \max(a1, a2)$$

$$result = sum - delta ,$$

где

$$delta = \begin{cases} 0, \text{ при } sum \bmod 3 \neq 0 \\ \min(\{d \mid d = |a1 - a2| : |a1 - a2| \bmod 3 \neq 0\}) \end{cases}$$



Пример

```
1  N = int(input())
2
3  sum = 0
4  delta = 0
5  for i in xrange(N):
6      a1, a2 = [int(x) for x in raw_input().split()]
7      sum += max(a1, a2)
8      if delta % 3 == 0:
9          delta = abs(a1-a2)
10     elif abs(a1-a2) % 3 > 0:
11         delta = min(delta, abs(a1-a2))
12
13  print sum if sum % 3 != 0 else sum - delta \
14      if sum - delta % 3 != 0 else 0
```

Спасибо за внимание!
Все вопросы по адресу:
AAMaksutov@mephi.ru

ИНФОРМАТИКА

Разбор прикладных задач

Максутов Артем Артурович,
Ассистент кафедры 12

Москва, 2017

Перевод чисел

Укажите наименьшее четырёхзначное шестнадцатеричное число, двоичная запись которого содержит ровно 6 нулей. В ответе запишите только само шестнадцатеричное число, основание системы счисления указывать не нужно.

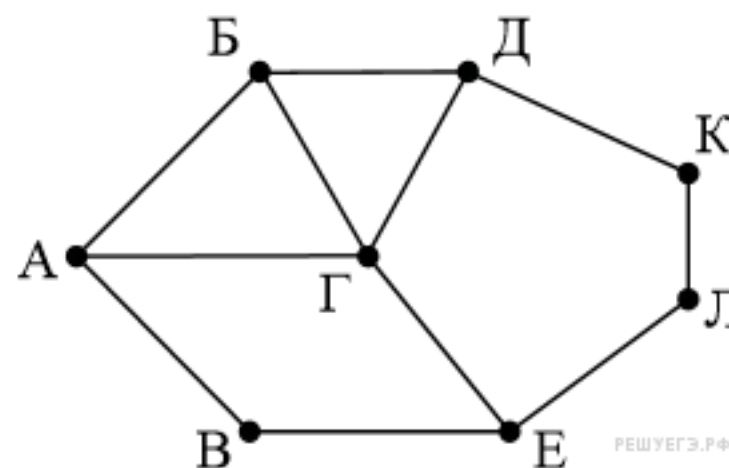
Перевод чисел

3	2	1	0	
			1	1
0	0	0	0	0
0	0	1	1	3
1	1	1	1	F

Графы

На рисунке схема дорог Н-ского района изображена в виде графа, в таблице содержатся сведения о длине этих дорог (в километрах).

	п1	п2	п3	п4	п5	п6	п7	п8
п1		15		20				18
п2	15		25					
п3		25				24		22
п4	20						12	
п5						13	16	17
п6			24		13			15
п7				12	16			
п8	18		22		17	15		



Так как таблицу и схему рисовали независимо друг от друга, нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите длину дороги из пункта Е в пункт Л. В ответе запишите целое число.



Автомат получает на вход четырёхзначное число (число не может начинаться с нуля). По этому числу строится новое число по следующим правилам.

1. Складываются отдельно первая и вторая, вторая и третья, третья и четвёртая цифры заданного числа.
2. Наименьшая из полученных трёх сумм удаляется.
3. Оставшиеся две суммы записываются друг за другом в порядке неубывания без разделителей.

Пример. Исходное число: 1984. Суммы: $1 + 9 = 10$, $9 + 8 = 17$, $8 + 4 = 12$. Удаляется 10. Результат: 1217.

Укажите **наибольшее** число, при обработке которого автомат выдаёт результат 613



- $6 \rightarrow 6+0, 5+1, 4+2, 3+3$
- $13 \rightarrow 9+4, 8+5, 7+6$
- 9424



Задача

В ответе не нужно перечислять все различные наборы значений перменной системы равенств. В качестве ответа Вам нужно указать количество

Вариант № 22419

1. Сколько существует различных наборов значений логических переменных, удовлетворяющих всем перечисленным ниже условиям?

$$(x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge x_2) \vee (x_2 \wedge x_3) \vee (\neg x_2 \wedge \neg x_3) = 1$$

$$(x_2 \wedge \neg x_3) \vee (\neg x_2 \wedge x_3) \vee (x_3 \wedge x_4) \vee (\neg x_3 \wedge \neg x_4) = 1$$

...

$$(x_8 \wedge \neg x_9) \vee (\neg x_8 \wedge x_9) \vee (x_9 \wedge x_{10}) \vee (\neg x_9 \wedge \neg x_{10}) = 1$$

В ответе не нужно перечислять все различные наборы значений переменных, удовлетворяющих данной системе равенств. В качестве ответа Вам нужно указать количество

1. Аналитический
2. Бинарный



Аналитический

- $(x_1 \& \overline{x_2}) \vee (\overline{x_1} \& x_2) \vee (x_2 \& x_3) \vee (\overline{x_2} \& \overline{x_3}) = 1$
- $(x_1 \oplus x_2) \vee (x_2 \equiv x_3) = 1$



Аналитический

- 0000000000
- 1111111111
- 0101010101
- 1010101010
- 01...11
- 10...00

Диаграмма Вейча

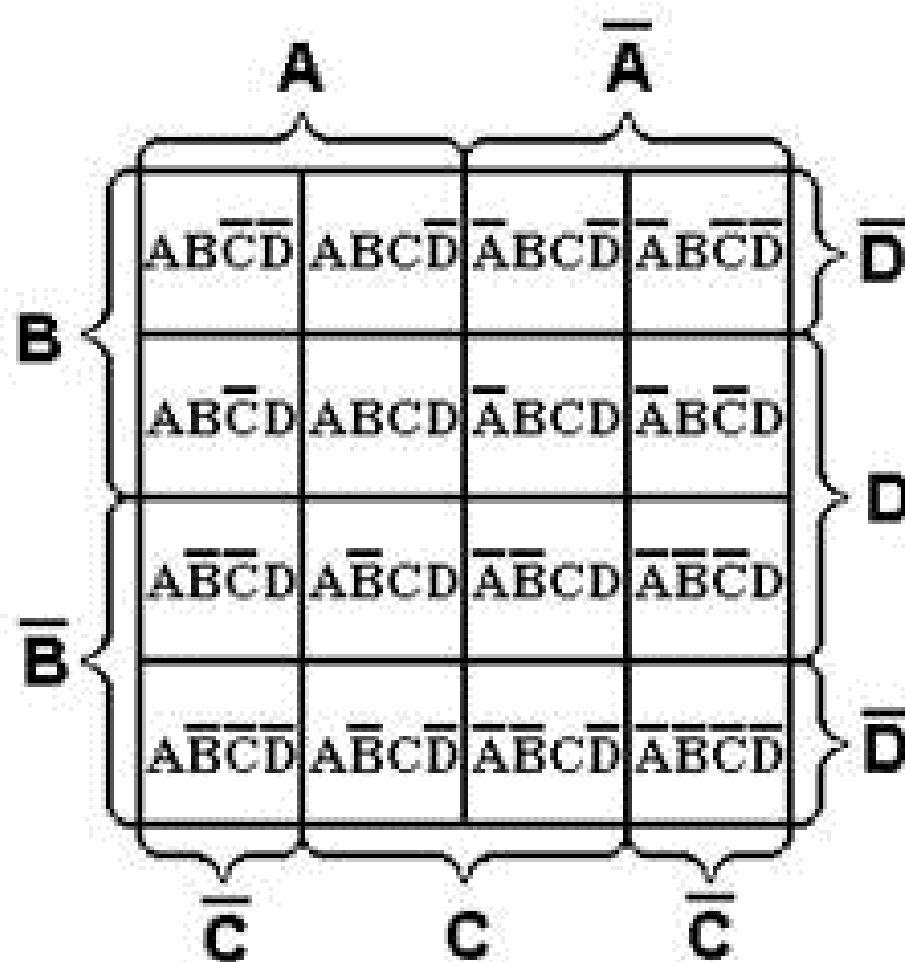
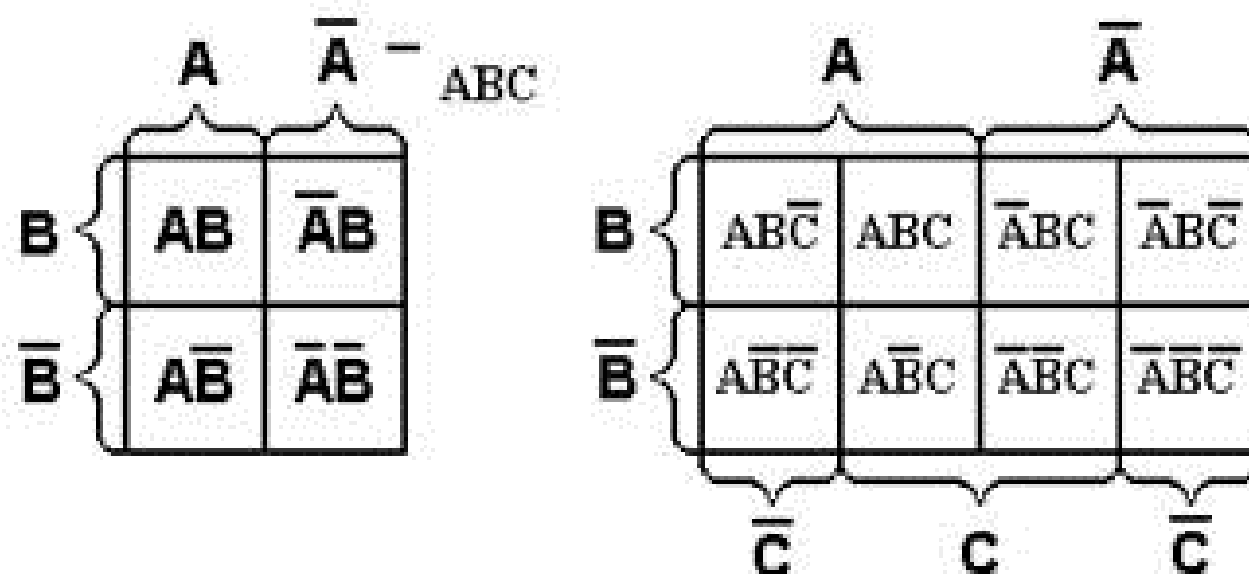


Диаграмма Вейча

	A		$\bar{A}$	
B	1	1	0	0
$\bar{B}$	0	0	0	0
	$\bar{C}$	C	$\bar{C}$	

$$f_3 = AB\bar{C} \vee ABC = AB(C \vee \bar{C}) = AB$$

	A		$\bar{A}$	
B	0	1	1	0
$\bar{B}$	0	1	1	0
	$\bar{C}$	C	$\bar{C}$	

$$f_4 = C$$

Бинарный

x_1	x_2	x_3	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Бинарный

x_1	x_2	x_3	x_4	F1	F2
0	0	0	0/1	1	1/0
0	0	1	0/1	0	x
0	1	0	0/1	1	1/1
0	1	1	0/1	1	0/1
1	0	0	0/1	1	1/0
1	0	1	0/1	1	1/1
1	1	0	0/1	0	x
1	1	1	0/1	1	0/1



Пример

$$(C < A \vee C < B) \wedge \neg(C+1 < A) \wedge \neg(\bar{C}+1 < B)$$

Чему равно C , если $A=45$ и $B=18$?

7. Сколько существует различных наборов значений логических переменных, удовлетворяющих всем перечисленным ниже условиям:

$$\neg((\neg x_1 \wedge x_2 \wedge \neg x_3) \vee (\neg x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \neg x_2 \wedge \neg x_3)) = 1,$$

$$\neg((\neg x_2 \wedge x_3 \wedge \neg x_4) \vee (\neg x_2 \wedge x_3 \wedge x_4) \vee (x_2 \wedge \neg x_3 \wedge \neg x_4)) = 1,$$

...

$$\neg((\neg x_8 \wedge x_9 \wedge \neg x_{10}) \vee (\neg x_8 \wedge x_9 \wedge x_{10}) \vee (x_8 \wedge \neg x_9 \wedge \neg x_{10})) = 1.$$

В ответе не нужно перечислять все различные наборы значений переменных, удовлетворяющих данной системе равенств. В качестве ответа Вам нужно указать количество таких наборов.

8. Сколько различных решений имеет система уравнений

Бинарный

x_1	x_2	x_3	F
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Бинарный

x_1	x_2	x_3	x_4	F1	F2
0	0	0	0/1	1	1/1
0	0	1	0/1	1	0/0
0	1	0	0/1	0	x
0	1	1	0/1	0	x
1	0	0	0/1	0	x
1	0	1	0/1	1	0/0
1	1	0	0/1	1	0/1
1	1	1	0/1	1	1/1

Спасибо за внимание!
Все вопросы по адресу:
AAMaksutov@mephi.ru