Введение в компьютерные науки

ЛЕКТОР К.Т.Н. МОХОВ В. А.

ГЛАВА 12. ТЕОРИЯ ВЫЧИСЛЕНИЙ

Глава12: Теория вычислений

- 12.1 Функции и их вычисление
- 12.2 Машины Тьюринга
- 12.3 Универсальные языки программирования
- 12.4 Невычислимые функции
- 12.5 Сложность задач
- 12.6 Криптография с использованием открытых ключей

Функции

- **Функция:** Соответствие между количеством входных и выходных значений набора двоичных разрядов , так чтоб на каждое возможное входное значение было назначено выходное .

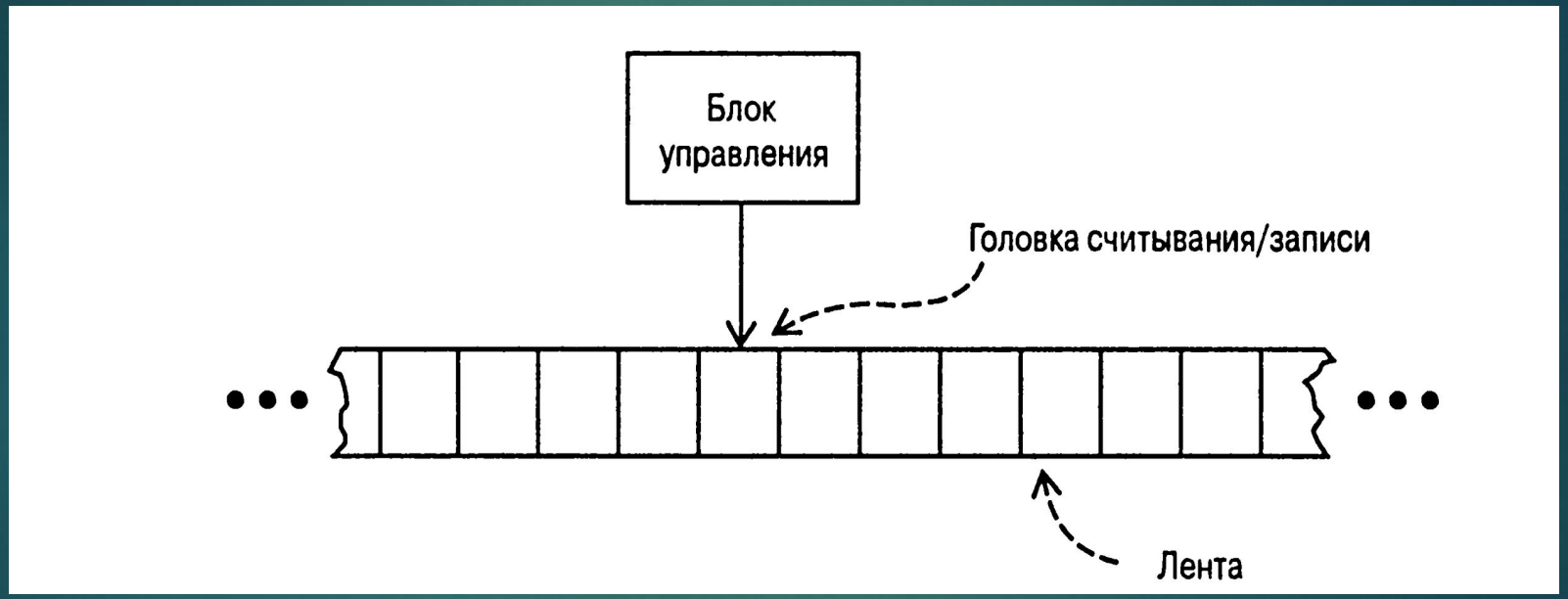
Функции (продолжение)

- **Вычисление ФУНКЦИИ:** Процесс определения выходной величины функции на основе значения ее входной величины
- **Невычислимая ФУНКЦИЯ:** Функция, которая не может быть вычислена по любому алгоритму

Рисунок 12.1 Попытка отобразить функцию, которая преобразует измерения в ярдах в метры

Ярды (вход)	Метры (выход)
1	0.9144
2	1.8288
3	2.7432
4	3.6576
5	4.5720
.	.
.	.
.	.

Рисунок 12.2 Компоненты Машины Тьюринга



Операции Машины Тьюринга

- Ввод данных на каждом шаге
  - Состояние
  - Значение по текущей позиции ленты
- Действия на каждом шаге
  - Запись значения в текущую позицию ленты
  - Чтение шагов /запись заголовков
  - Смена состояния

Рисунок 12.3 Состояние Машины Тьюринга предназначенной для увеличения числа

Текущее состояние	Содержание текущей ячейки	Записываемое значение	Направление перемещения	Следующее состояние
START	*	*	Влево	ADD
ADD	0	1	Влево	NO CARRY
ADD	1	0	Влево	CARRY
ADD	*	*	Вправо	HALT
CARRY	0	1	Влево	NO CARRY
CARRY	1	0	Влево	CARRY
CARRY	*	1	Влево	OVERFLOW
NO CARRY	0	0	Влево	NO CARRY
NO CARRY	1	1	Влево	NO CARRY
NO CARRY	*	*	Вправо	RETURN
OVERFLOW	Игнорируется	*	Вправо	RETURN
RETURN	0	0	Вправо	RETURN
RETURN	1	1	Вправо	RETURN
RETURN	*	*	Нет перемещения	HALT

Тезис Черча-Тьюринга

- Любая функция, которая может быть вычислена физическим устройством, может быть вычислена машиной Тьюринга

Универсальный язык программирования

Язык, которым может быть выражено решение любой вычислимой функции

- Примеры: “Bare Bones” и самые популярные языки программирования

Язык Bare Bones

- Bare Bones это простой , но универсальный язык.
- Операторы
  - **clear *name*;**
  - **incr *name*;**
  - **decr *name*;**
  - **while *name* not 0 do; ... end;**

Рисунок 12.4 Программа для вычисления X и Y на Bare Bones

```
clear Z;
while X not 0 do;
    clear W;
    while Y not 0 do;
        incr Z;
        incr W;
        decr Y;
    end;
    while W not 0 do;
        incr Y;
        decr W;
    end;
    decr X;
end;
```

Рисунок 12.5 Выполнение инструкции «copy Today to Tomorrow» на Bare Bones

```
clear Aux;
clear Tomorrow;
while Today not 0 do;
    incr Aux;
    decr Today;
end;
while Aux not 0 do;
    incr Today;
    incr Tomorrow;
    decr Aux;
end;
```

Проблема остановки

- Учитывая кодированную версию любой программы, возвращает 1, если программа заканчиваются автоматически, или 0, если программа не является таковой.

Рисунок 12.6 Тестирование программы само завершения

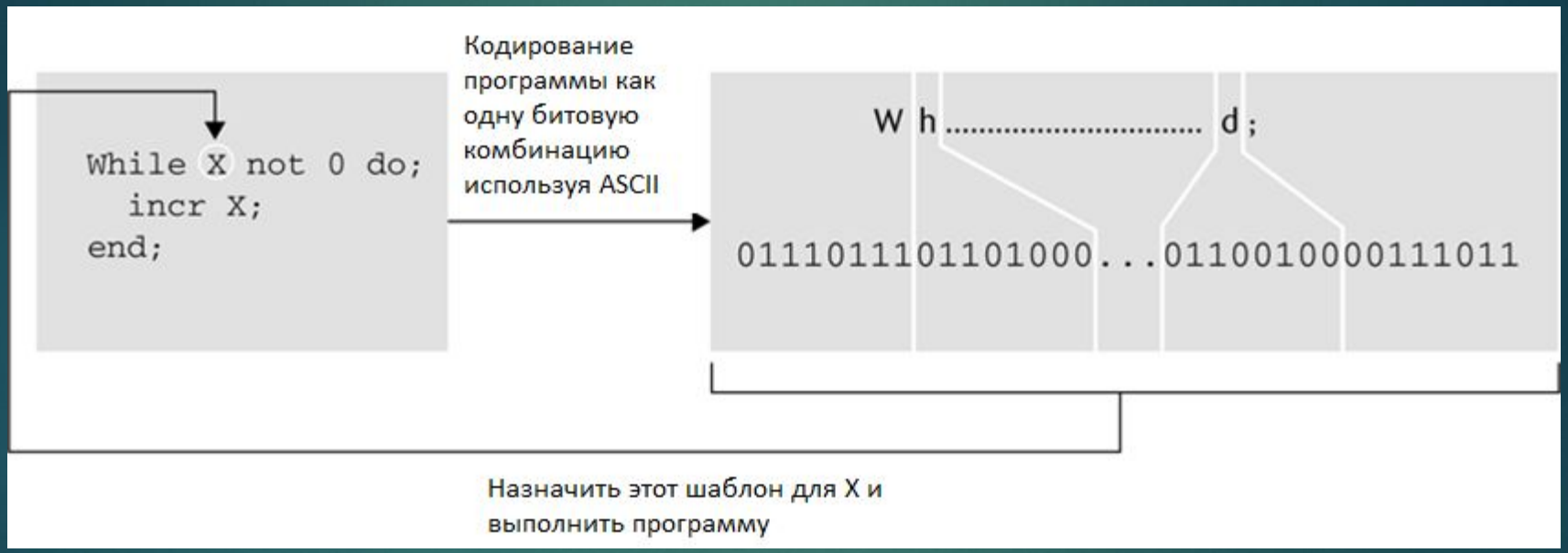
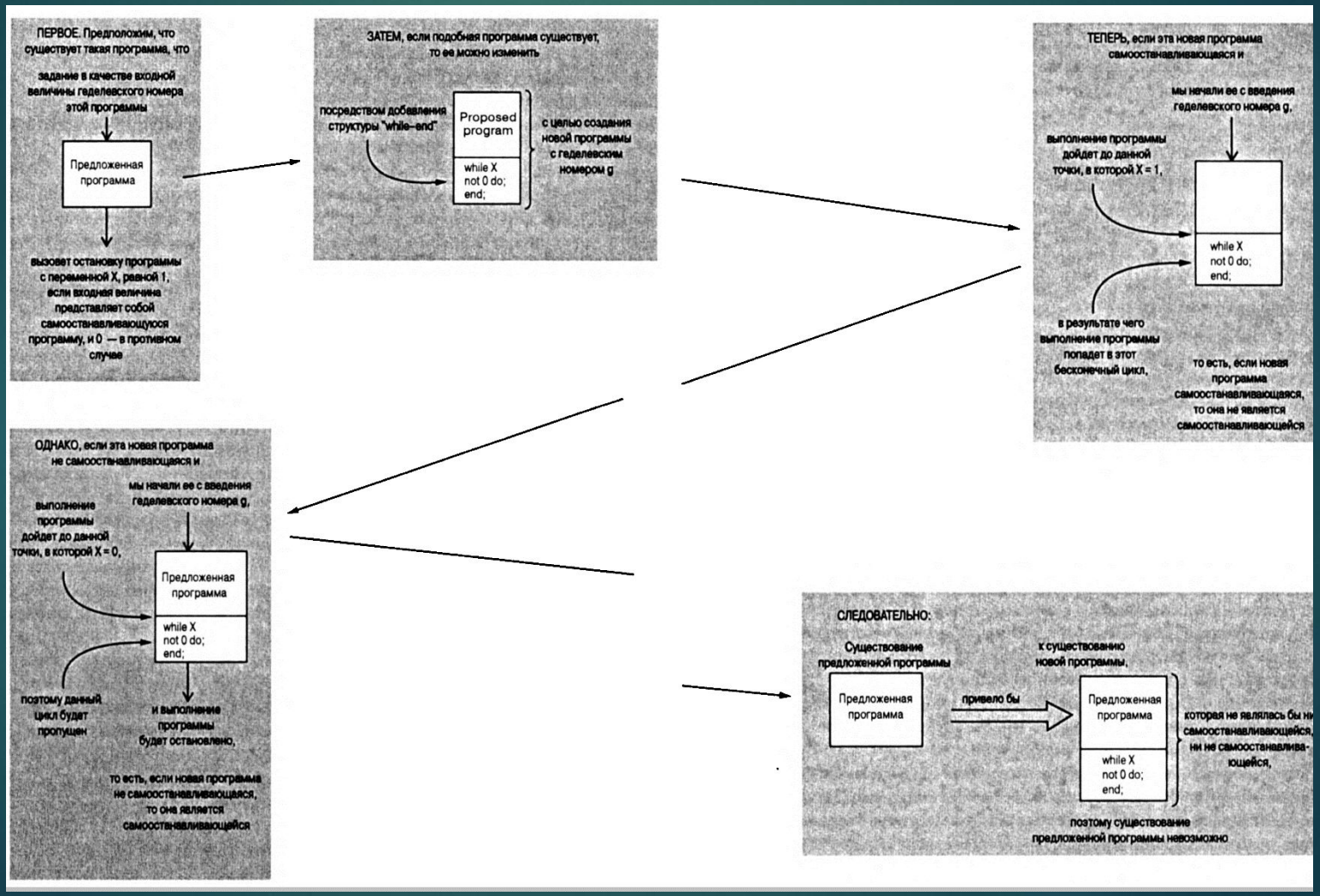


Рисунок 12.7 Доказательство неразрешимости проблемы остановки программным путем



Сложность задач

- **Время сложности:** Количество требуемых для исполнения команд
  - Если не указано иное «сложность» означает «время сложности»
- Если алгоритм класса 0(lgn) более эффективен, чем алгоритм класса 0(n), то алгоритм класса 0(n) является более сложным, чем алгоритм класса 0(lgn).

- То, что задача принадлежит к классу O(f(n)), равносильно утверждению о существовании ее решения (не обязательно лучшего), сложность которого равна 0(f(n)).

Рисунок 12.8 Процедура MergeLists для слияний двух списков

```
procedure MergeLists (<входной список A>, <входной список B>,
                      <выходной список>)
if (оба входных списка пусты) then (завершить работу, причем <выходной список> пуст)
if (<входной список A> пуст)
  then (объявить его оконченным)
  else (объявить его первый элемент текущим)
if (<входной список B> пуст)
  then (объявить его оконченным)
  else (объявить его первый элемент текущим)
while (ни один из входных списков не окончен) do
  (поместить в <выходной список> текущую запись с "меньшим" значением
      поля ключа;
   if (эта текущая запись является последней в соответствующем
      входном списке)
    then (объявить этот входной список оконченным)
    else (объявить следующую запись этого входного списка текущей)
  )
Начиная с текущей записи списка, который еще не окончен, копировать
 оставшиеся записи в <выходной список>.
```

Рисунок 12.9 Алгоритмы сортировки слиянием реализованный в виде процедуры MergeSort

```
procedure MergeSort (<список>)
 if (<список> имеет более одного элемента)
   then (вызвать процедуру MergeSort для сортировки первой
         половины списка;
        вызвать процедуру MergeSort для сортировки второй
         половины списка;
        вызвать процедуру MergeLists для слияния первой и второй
        половин списка с получением полностью отсортированной
        версии исходного списка
        )
```

Рисунок 12.10 Иерархическое представление множества задач порожденных алгоритмом сортировки методом слияния

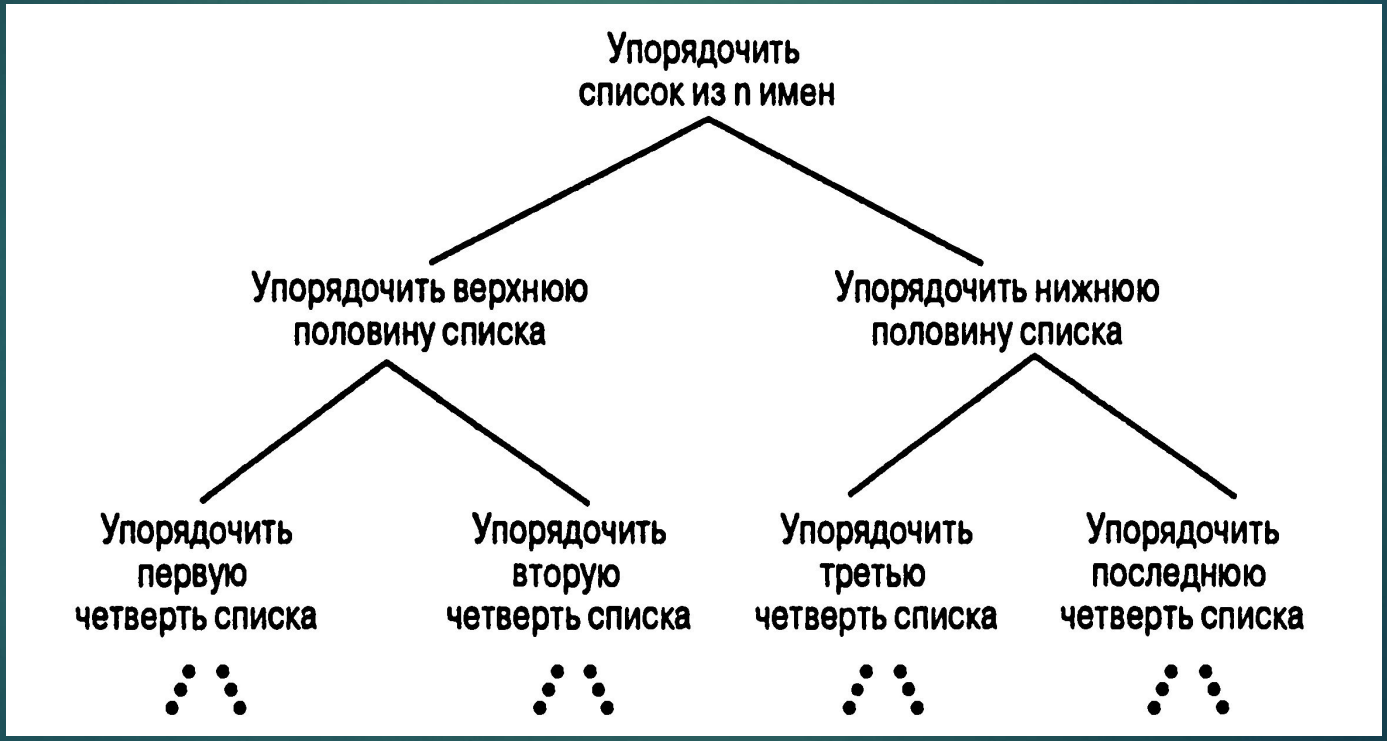
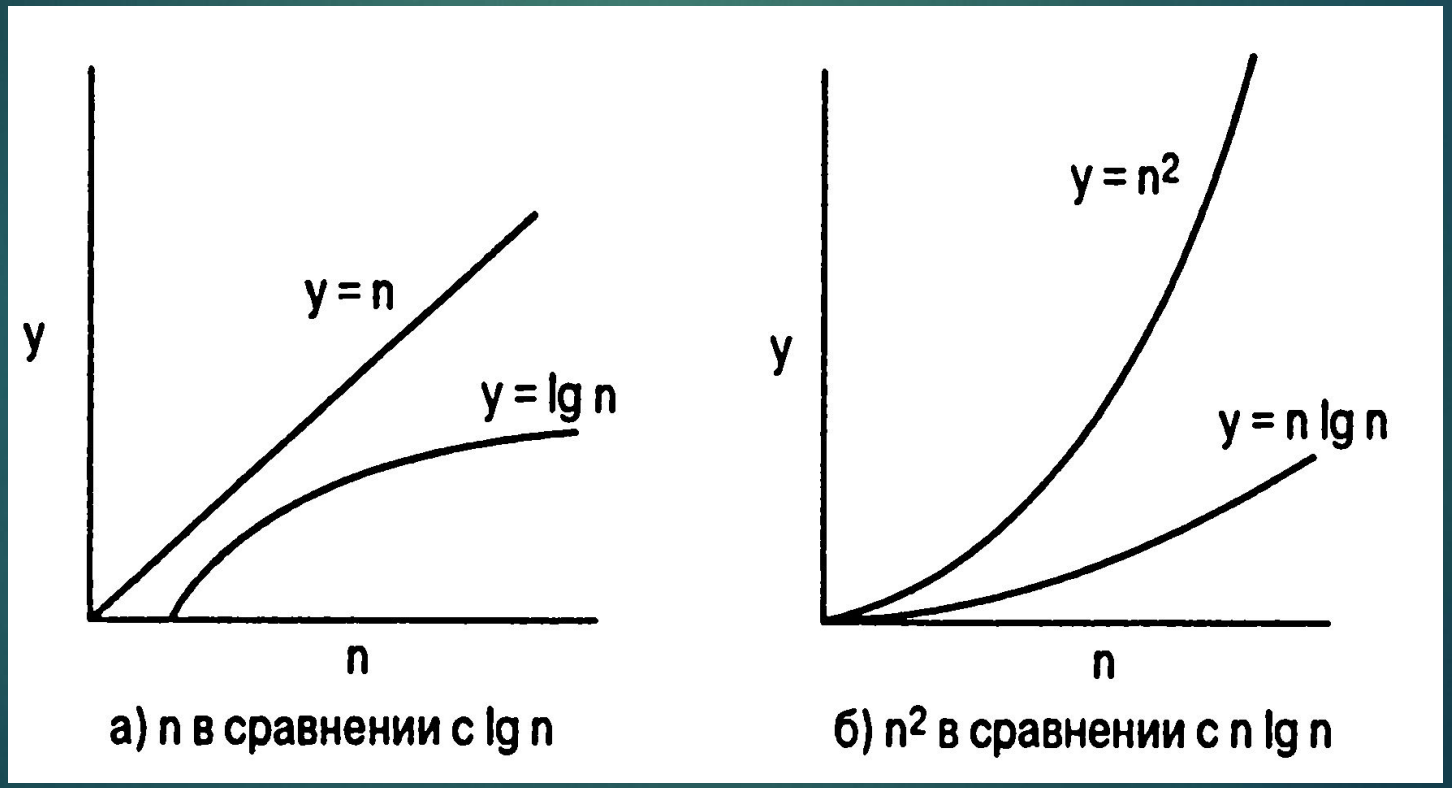


Рисунок 12.11 График основных типов математических функций



а) n в сравнении с lg n

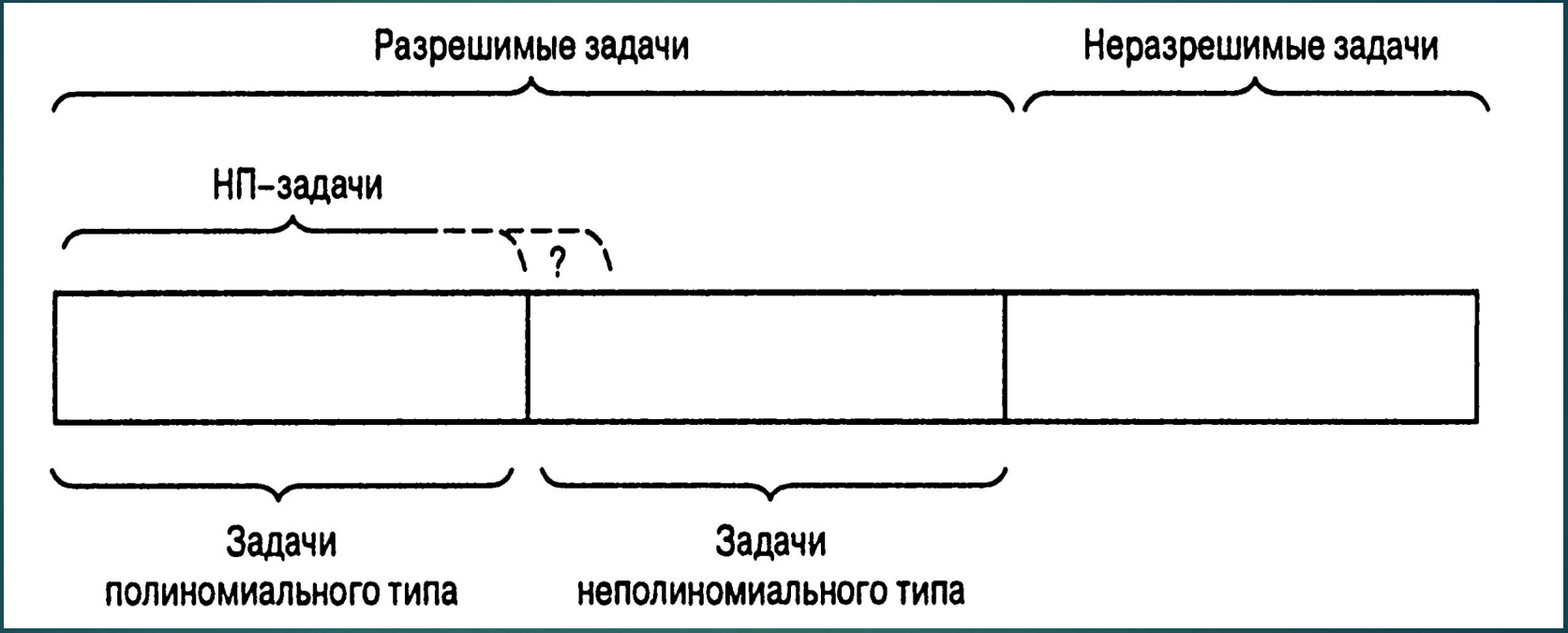
б) n^2 в сравнении с n lg n

P против NP

- **Класс P:** Задача в классе $\Theta(f(n))$, где $f(n)$ является полиномом
- **Класс NP:** Все задачи могут быть решены в недетерминированным алгоритмом в полиноминальное время

  Недетерминированный алгоритм = “алгоритм”, шаги которого не могут быть однозначно и полностью определены состоянием процесса
- Больше ли класс NP чем класс P в настоящее время неизвестно

Рисунок 12.12 Графическое обобщение классификации задач



Криптография с использованием открытых ключей

- **Ключ:** Значение используемое для шифровке и дешифровке сообщения
  - **Открытый ключ**: Используется для шифровки сообщений
  - **Секретный ключ**: Используется для дешифровки сообщения
- **RSA:** Популярный криптографический алгоритм с открытым ключом
  - Опирается на (предполагаемую) неподатливость проблемы разложения больших чисел на множители

Шифрование сообщения 10111

- Шифрование ключей: n = 91 и e = 5
- $10111_2 = 23_{10}$
- $23^e = 23^5 = 6{,}436{,}343$
- 6,436,343 ÷ 91 имеет остаток от 4
- $4_{10} = 100_2$
- Таким образом, зашифрованная версия 10111 равняется100.

Дешифровка сообщения 100

- Расшифровка ключей: d = 29, n = 91
- $100_2 = 4_{10}$
- $4^d = 4^{29} = 288{,}230{,}376{,}151{,}711{,}744$
- 288,230,376,151,711,744 ÷ 91 имеет остаток 23
- $23_{10} = 10111_2$
- Таким образом расшифрованная версия 100 является 10111.

Рисунок 12.13 Шифрование с использованием открытого ключа

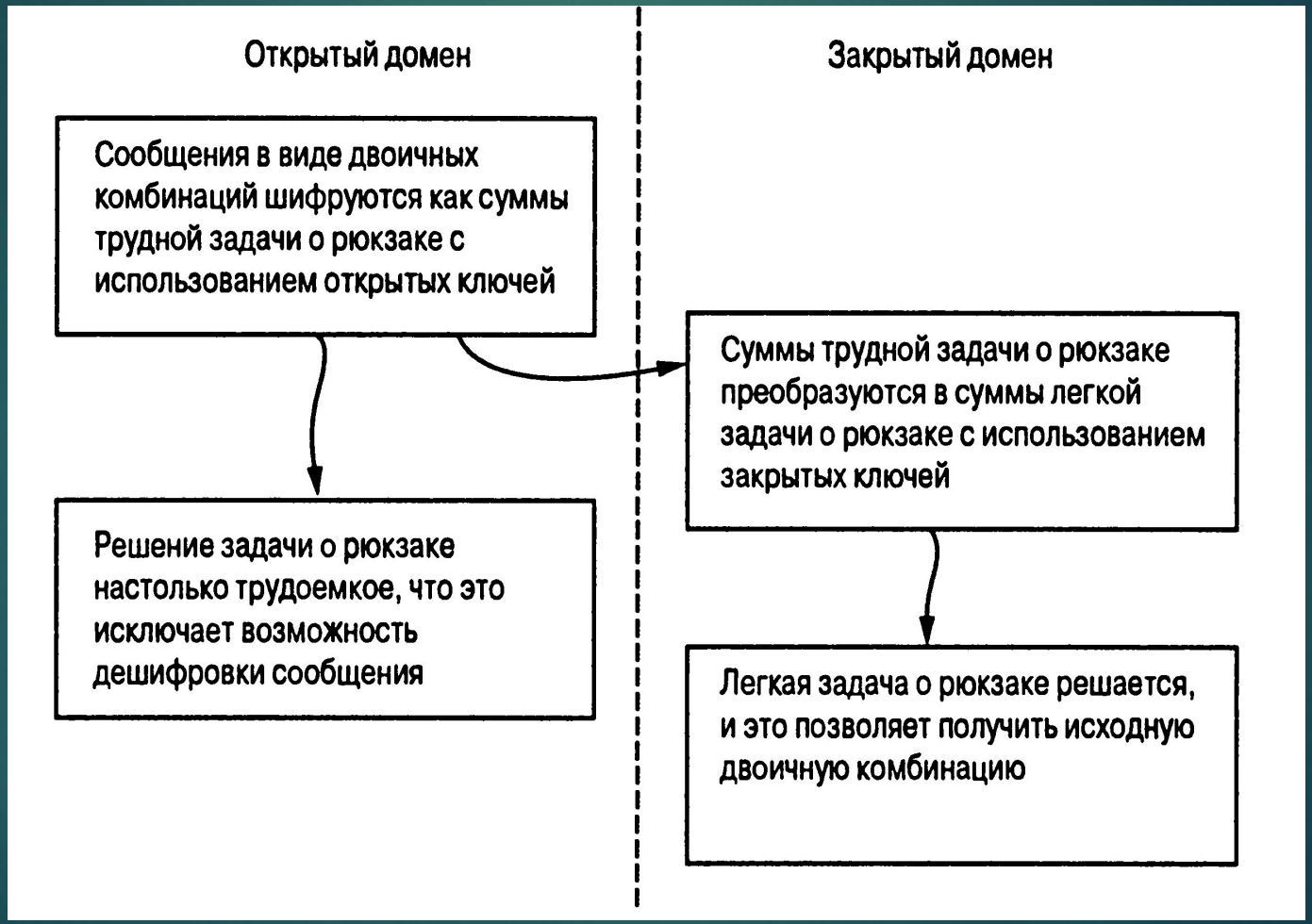


Рисунок 12.14 Установка системы шифрования открытого ключа RSA