

# **Тема 5.1. Алгоритмы и структуры данных.**

# Роль алгоритмов и структур данных



## Понятие алгоритма

*Алгоритм — это конечный набор правил, который определяет последовательность операций для решения конкретного множества задач и обладает пятью важными чертами: конечность, определённость, ввод, вывод, эффективность.*



Дональд Эрвин Кнут

## Свойства алгоритма

**Дискретность** — алгоритм должен представлять процесс решения задачи как последовательное выполнение некоторых простых шагов. При этом для выполнения каждого шага алгоритма требуется конечный отрезок времени.

**Детерминированность** — определённость. В каждый момент времени следующий шаг работы однозначно определяется состоянием системы. Таким образом, алгоритм должен выдавать один и тот же результат для одних и тех же исходных данных.

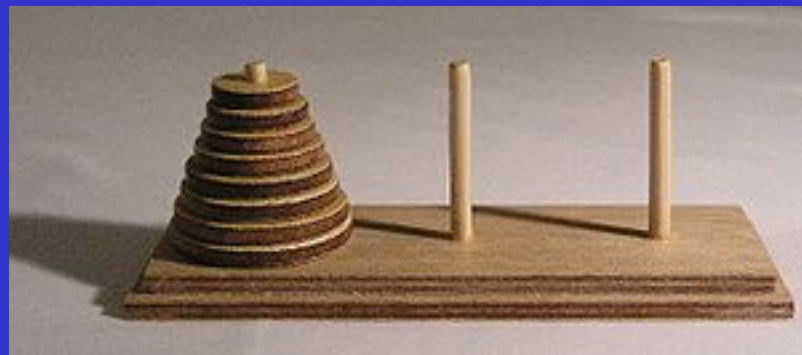
**Конечность** — при корректно заданных исходных данных алгоритм должен завершать работу и выдавать результат за конечное число шагов.

**Масштабируемость** — алгоритм должен быть применим к разным наборам исходных данных.

**Результативность** — завершение алгоритма определенными результатами.

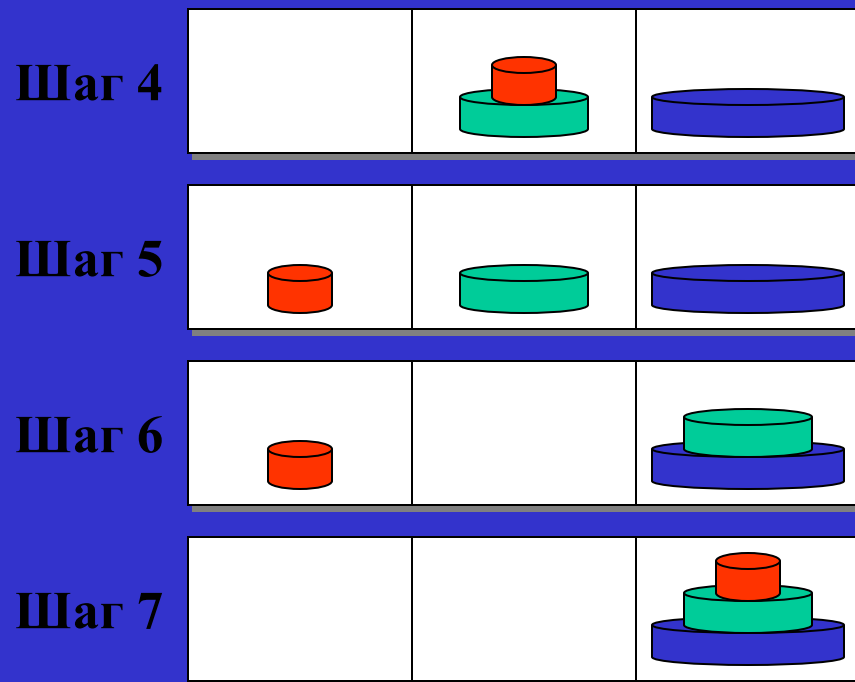
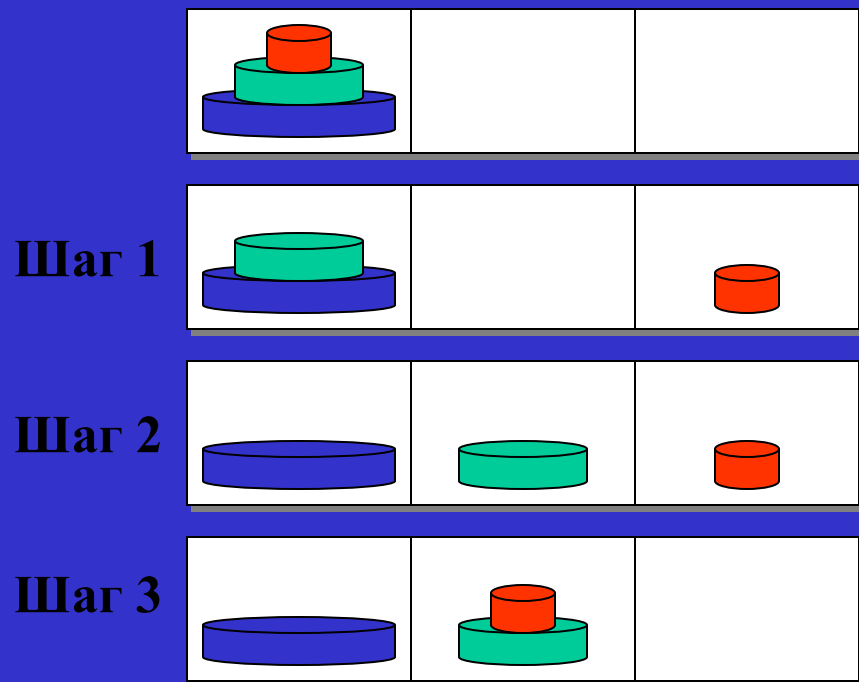
**Эффективность** — завершение алгоритма определенными результатами за определенное число шагов (время).

## Пример алгоритма - задача о Ханойских башнях



**Легенда.** В одном из буддийских монастырей монахи уже тысячу лет занимаются перекладыванием колец. Они располагают тремя пирамидами, на которых надеты кольца разных размеров. В начальном состоянии 64 кольца были надеты на первую пирамиду и упорядочены по размеру. Монахи должны переложить все кольца с первой пирамиды на вторую, выполняя единственное условие — кольцо нельзя положить на кольцо меньшего размера. При перекладывании можно использовать все три пирамиды. Монахи перекладывают одно кольцо за одну секунду. Как только они закончат свою работу, наступит конец света...

## Решение задачи о Ханойских башнях



Число шагов алгоритма вычисляется по формуле  $2^N - 1$ , где  $N$  – число колец .

Для перекладывания 64-х колец потребуется **18 446 744 073 709 551 615** шагов.

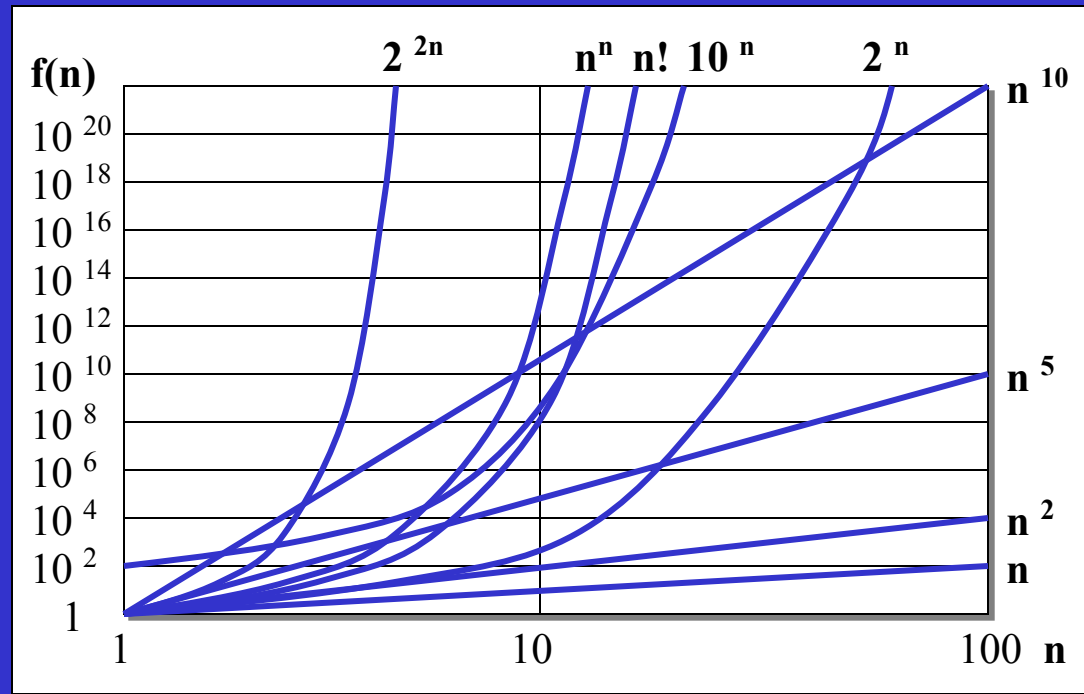
При скорости в одно перекладывание в секунду, потребуется около **584 542 046 091** лет.

# Эффективность алгоритмов

Временные функции  
сложности

Полиномиальные  
(P-задачи)

Экспоненциальные  
(NP-задачи)



## Трансвычислительные задачи

*Не существует системы обработки данных, искусственной или естественной, которая могла бы обрабатывать более  $2 \cdot 10^{47}$  бит в секунду на грамм своей массы.*

Ханс Бреммерман



**Предел Бреммермана**

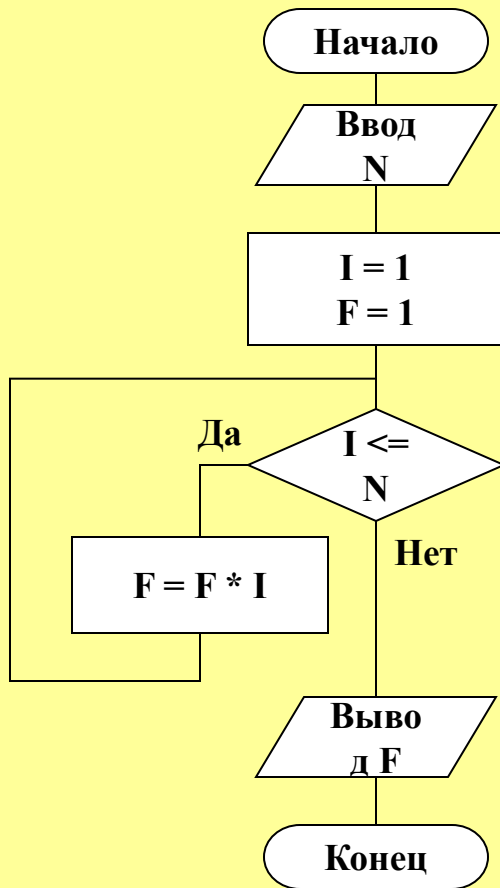
**$10^{93}$  бит**

**Трансвычислительные задачи**



## Представления алгоритмов

### Блок-схема



### Псевдокод

**Ввод N**

**I = 1**

**F = 1**

**ЦИКЛ ПОКА I <= N**








**F = F \* I**

**ВСЁ-ЦИКЛ**

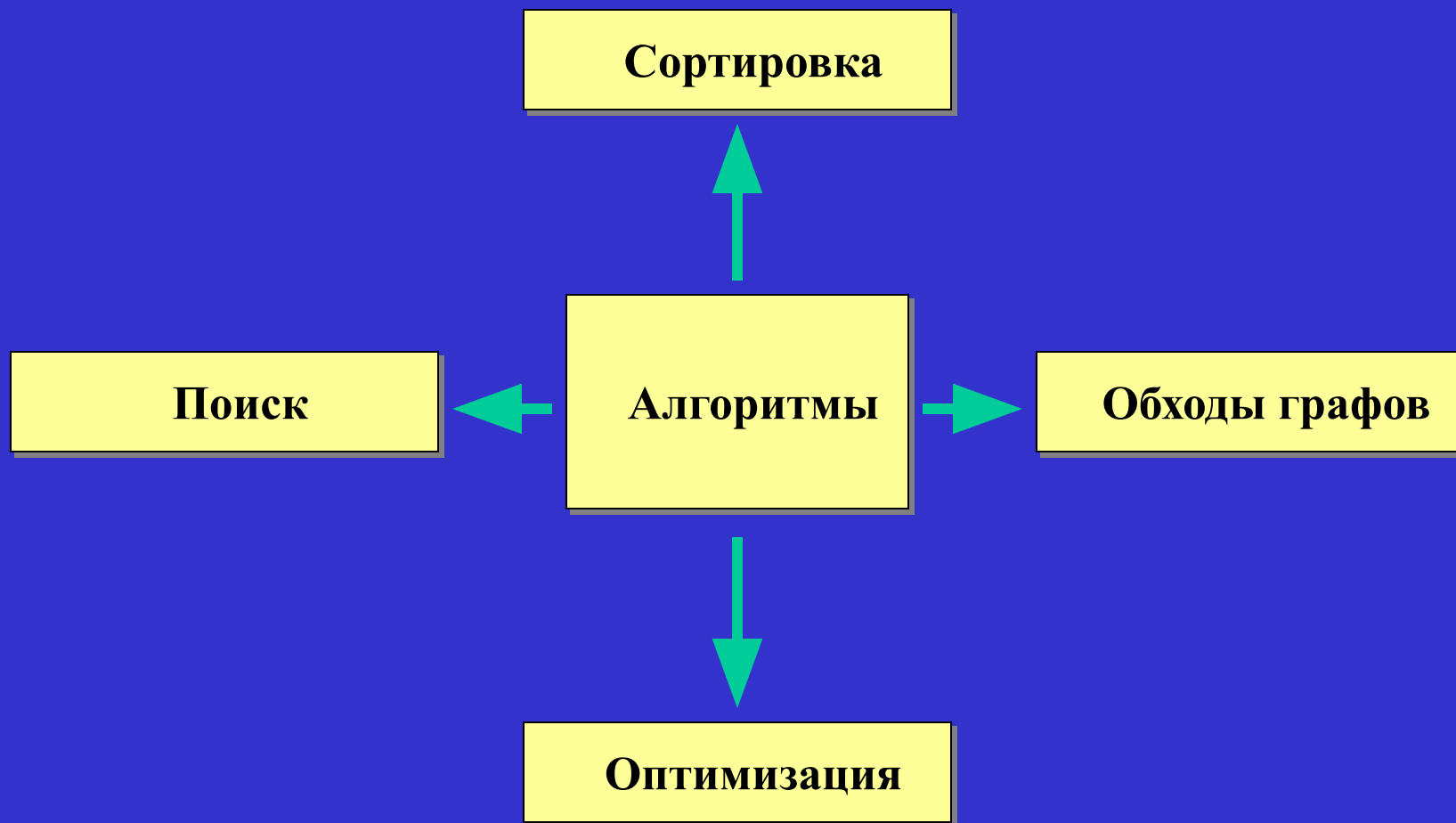
**Вывод F**

## Блок-схемы алгоритмов

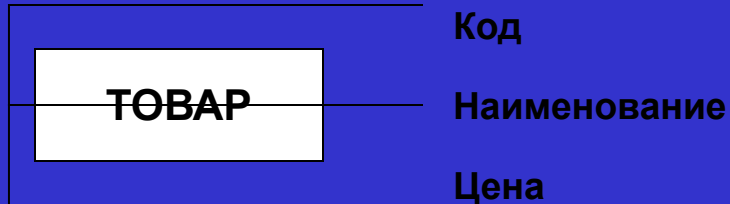
ГОСТ 19.701-90 (ИСО 5807-85). Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.

Наименование	Обозначение	Функция
<b>Терминатор</b>		Элемент отображает вход из внешней среды или выход из нее (наиболее частое применение – начало и конец программы).
<b>Процесс</b>		Выполнение одной или нескольких операций, обработка данных любого вида
<b>Решение</b>		Отображает решение с одним входом и двумя или более альтернативными выходами, из которых только один может быть выбран.
<b>Предопределенный процесс</b>		Символ отображает выполнение процесса, который определен в другом месте программы
<b>Данные</b>		Преобразование данных в форму, пригодную для обработки (ввод) или отображения результатов обработки (вывод).
<b>Соединитель</b>		Символ отображает выход в часть схемы и вход из другой части этой схемы.
<b>Комментарий</b>		Используется для более подробного описания шага, процесса или группы процессов.

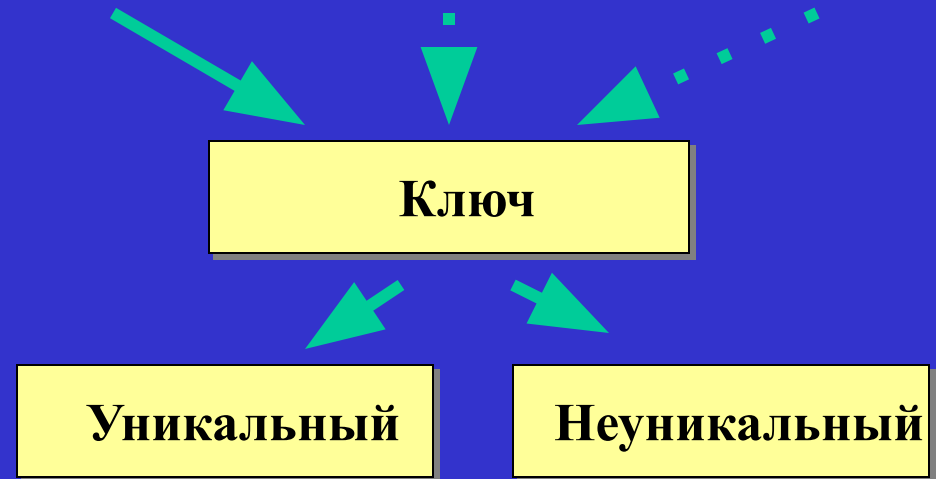
## Классы алгоритмов



## Сортировка массивов



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



*Сортировка - упорядочение массива в соответствии со значениями ключа*

# Алгоритмы сортировки массивов

Сортировка

Сортировка с помощью включения

Сортировка с помощью выделения

Сортировка с помощью обмена

$n^2$

Прямое включение

Двоичное включение

Прямой выбор

Пузырьковая

Шейкерная

Прямые

$n \cdot \log(n)$

Включение с уменьшающимися расстояниями (Шелла)

С помощью дерева

Разделение (быстрая)

Улучшенные