

# Алгоритмы и структуры данных

Lecture Notes N 6 (v.4)

Б. Мишнев



# Определение

- **Алгоритм** – это упорядоченный набор из недвусмысленных и выполнимых этапов, определяющий некоторый конечный процесс
  - **Упорядоченный**
  - **Осуществимый (эффективный)**
  - **Недвусмысленный**
  - **Конечный**

# Викторина 1

- Нарисуйте в конспекте таблицу

№ вопр.	Ответ	Проверка
1.1		
1.2		
1.3		
1.4		
1.5		

- В качестве ответов используйте:  
I – да, - нет, O – не известно

## Вопрос 1.1

- Верно ли, что один алгоритм может описывать несколько последовательностей выполнения отдельных этапов, выполняемых одновременно ?

## Вопрос 1.2

- Является ли алгоритмом запись следующей алгебраической формулы?

$$F = (9 / 5) * C + 32$$

## Вопрос 1.3

- Является ли нижеследующая последовательность инструкций алгоритмом?
  1. Составить список всех целых чисел
  2. Упорядочить список по убыванию
  3. Выбрать первое по порядку число из этого списка

## Вопрос 1.4

- Верно ли, что программа является алгоритмом решения задачи?

## Вопрос 1.5

- Верно ли, что описанный в литературе Буриданов осел погиб из-за «неопределенности / двусмысленности» своего алгоритма поведения?



# Викторина 1 (ответы)

- Сравните свои ответы и отметьте правильные в графе проверка «галочкой»

№ вопроса	Ответ	Проверка
1.1	I	
1.2	I	
1.3	-	
1.4	-	
1.5	I	

- Количество правильных ответов запишите!

# Примитивы

- Составные блоки для конструирования представления алгоритмов - примитивы
- Синтаксис – относится к символьному представлению примитива
- Семантика – к значению примитива

# Язык программирования

- Набор примитивов вместе с набором правил, устанавливающих, как эти примитивы могут комбинироваться для представления более сложных идей, образуют язык программирования

# Уровень детализации

- На уровне команд – для выполнения машиной (машинный язык)
- На более высоком уровне – для разработки алгоритма (схемы алгоритмов, псевдокод)

# Викторина 2

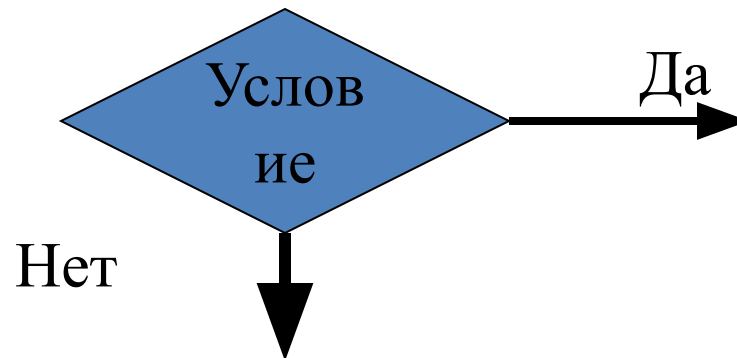
- Нарисуйте в конспекте таблицу

№ вопр.	Ответ	Проверка
2.1		
2.2		
2.3		
2.4		
2.5		

- В качестве ответов используйте:  
I – да, - нет, O – не известно

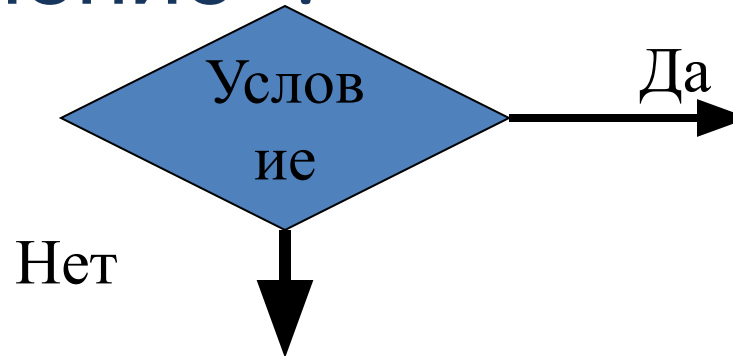
## Вопрос 2.1

- Правильно ли то, что семантическое значение примитива ниже – это «ромб»?



## Вопрос 2.2

- Правильно ли то, что синтаксическое значение примитива ниже – это «разветвление»?



## Вопрос 2.3

- Верно ли, что «псевдокод» можно получить путем ослабления правил того языка программирования, на котором требуется записать окончательную версию алгоритма?



## Вопрос 2.4

- Верно ли, что исторически самым распространенным способом записи алгоритмом при их проектировании являлось рисование «потоковых диаграмм» (flow charts)?

## Вопрос 2.5

- Верно ли, что существуют системы проектирования программного обеспечения, которые позволяют «исполнять» графическую запись алгоритма сразу без необходимости записи его на символьном языке программирования (например, на языке Pascal или Java)?

## Викторина 2 (ответы)

- Сравните свои ответы и отметьте правильные в графе проверка «галочкой»

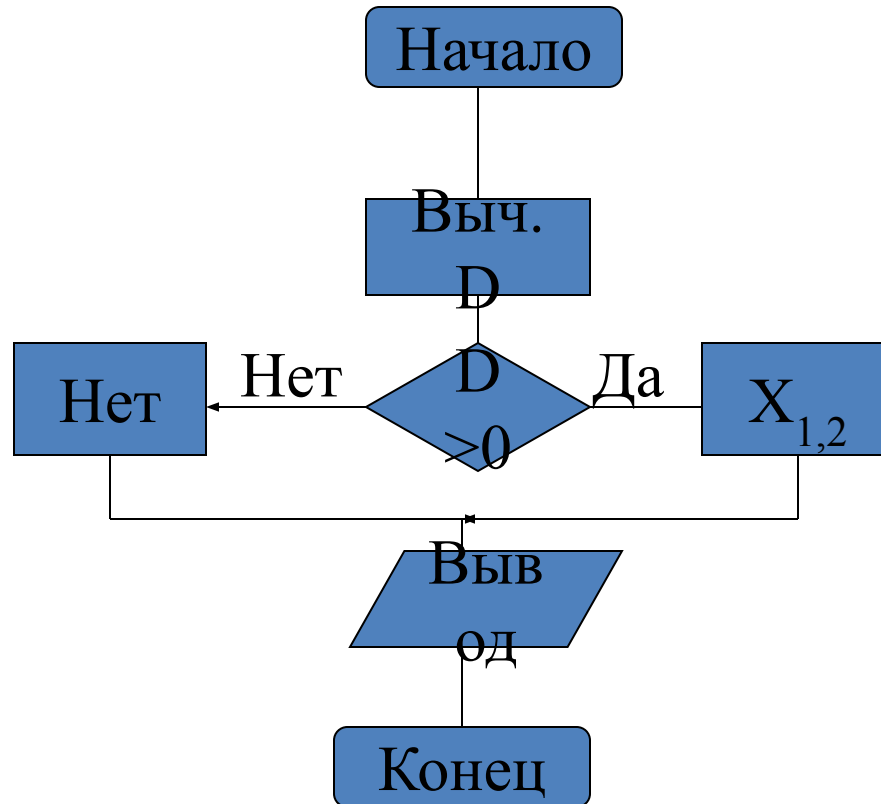
№ вопроса	Ответ	Проверка
2.1	-	
2.2	-	
2.3	I	
2.4	I	
2.5	I	

- Количество правильных ответов запишите!

# Типы алгоритмов

- Линейные
- Разветвляющиеся
- Циклические

# Пример схемы алгоритма



# Пример псевдокода

- начало
- Вычисл.  $D$
- если  $D > 0$ 
  - то Вычисл.  $X_{1,2}$
  - иначе Корней нет
- Вывести результат
- конец

## Метод игры в шахматы “brute force” (полный перебор).

- Из текущей позиции определяются все возможные ходы (около 60)
- Затем выбирается очередной ход из нерассмотренных на данном уровне, совершается, опять определяются все возможные ходы (уже за другую сторону), снова выбирается очередной ход, и так происходит до достижения некоторой заданной глубины.
- Из всех позиций, перебирающихся на заданной глубине, выбирается одна с максимальной оценкой, и первый ход, ведущий по ветке в эту позицию, объявляется лучшим.

# Притча Э. Дейкстры (E.W.Dijkstra, 1930-2002)

- Первый программист работал в времена первых железнодорожных компаний
- В целях экономии туалетами снабжали каждый второй железнодорожный вагон
- При комплектовании поездов отдельными вагонами возникало множество проблем
- Решение проблемы – в комплектовании поездов «сцепкой» из двух вагонов
- **Вывод: просто программировать мало, надо принимать во внимание и структуры данных**



# Professor Edsger Wybe Dijkstra



Dijkstra worked at the Mathematisch Centrum (MC/CWI) between 1952 and 1962 in Amsterdam. Perhaps his greatest achievement during these years was the writing with Jaap Zonneveld of the world's **first ALGOL60 compiler**.

From 1962 until 1984 he held a chair at the Eindhoven University of Technology. His fame was augmented through his fundamental studies of parallel programming and his insights into the construction of correct programs, and he was an eloquent advocate of the **methodology of structured programming**. From 1984 until his retirement in 1999 he worked at the University of Texas in Austin. Dijkstra was the 1972 recipient of the ACM **Turing Award**, often viewed as the Nobel Prize for computing. He wrote over 1300 books and papers, all of which are digitally accessible.

ALGORITHMS +  
DATA STRUCTURES =  
PROGRAMS

# Niklaus Wirth

- is a Swiss computer scientist,
- the chief designer of the programming languages Algol W, Pascal, Modula, Modula-2, and Oberon.
- He wrote *Algorithms + Data Structures = Programs*,
- He received the *Turing Award*



# Концепция типа данных

Каждая константа, переменная, выражение или функция имеет определенный **тип данных**, который определяет множество возможных значений

# Концепция типа для языка Pascal

- Тип данных определяет множество значений для константы, переменной, выражения и функции
- Тип данных может быть задан как контекстно (формой записи), так и специальным объявлением типа
- Каждый оператор или функция предполагает как аргумент определенного типа, так и результат определенного типа

# Примеры объявления типов

- `const n=10;`  
    `eps=0.001;`
- `type digit = 0..9;`
- `var i, j, r: integer;`  
    `a, b, c: real;`  
    `d: digit;`

# Стандартные типы данных

- **Скалярные типы** (integer, Boolean, char, real)
- **Сложные типы** (array, set, record)
- **Последовательный файл**
- **Текстовый файл**

# Динамические информационные структуры

- **Указатели (или ссылки)**
- **Линейные списки**
- **Древовидные структуры  
(деревья)**

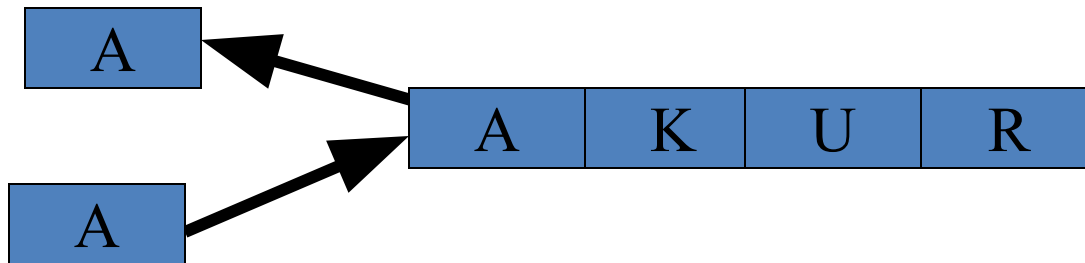
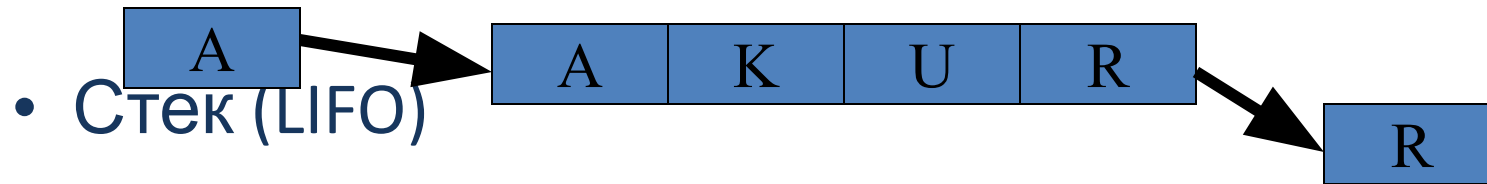


## Ссылка (Reference)

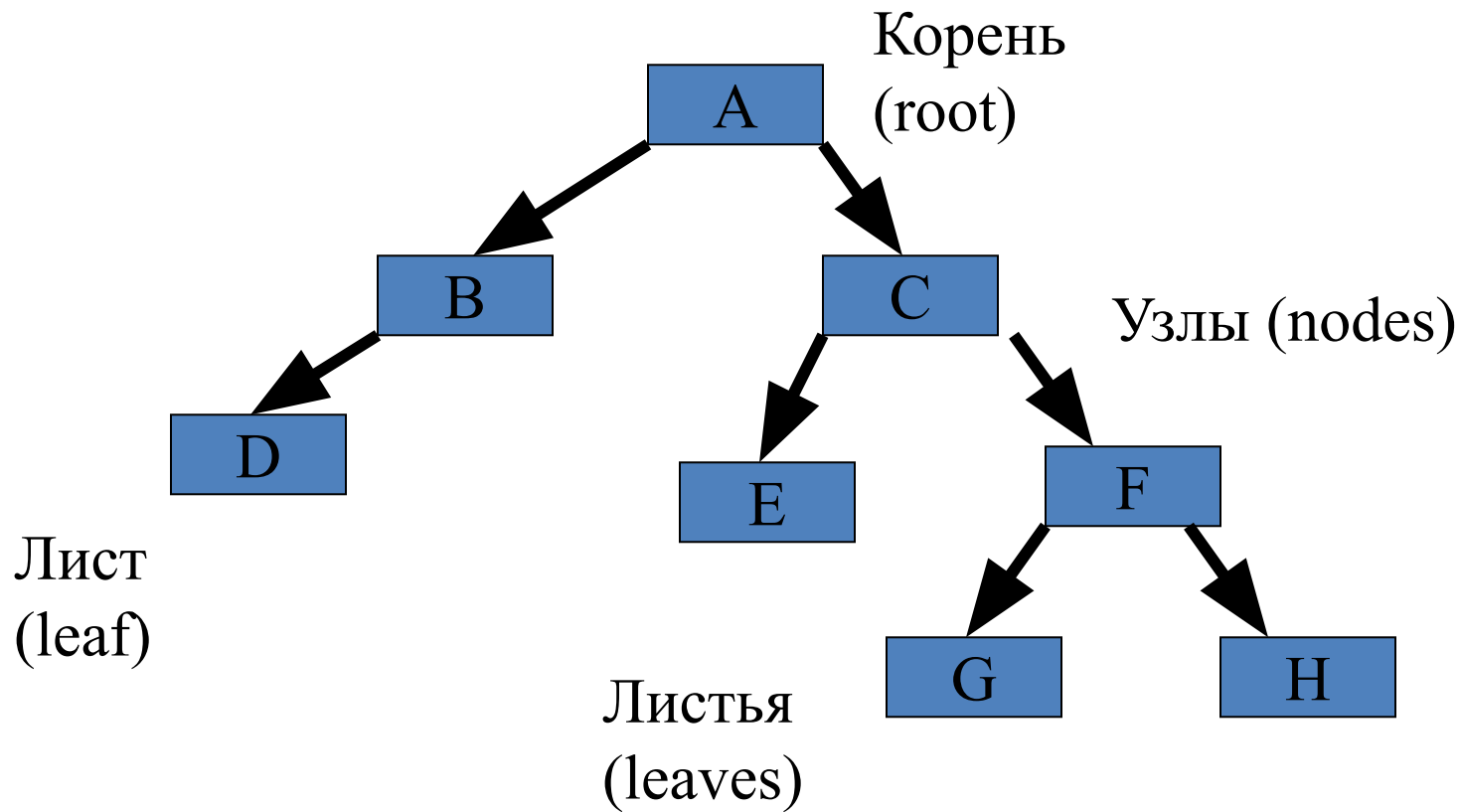
- Часть объекта, предназначенная для указания местонахождения другого объекта
- Имя объекта в некотором программном контексте

# Линейные списки

- Очередь (FIFO)



# Бинарное дерево (binary tree)



# Алгоритм последовательного поиска

- начало
- если список пуст то Поиск неудачный  
иначе  
выбрать первый элемент списка  
пока искомое значение не равно  
выбранному  
и есть непроверенные элементы  
выполнять выбрать следующий элемент  
если выбранное значение = искомому  
тогда Поиск успешный  
иначе Поиск неудачный
- конец

# Сортировки

- **Сортировки массивов**
  - Сортировка вставкой
  - Сортировка выбором
  - Сортировка обменом («пузырек»)
  - «Быстрая сортировка» (Quicksort)
- **Сортировки последовательных файлов**

# Викторина 3

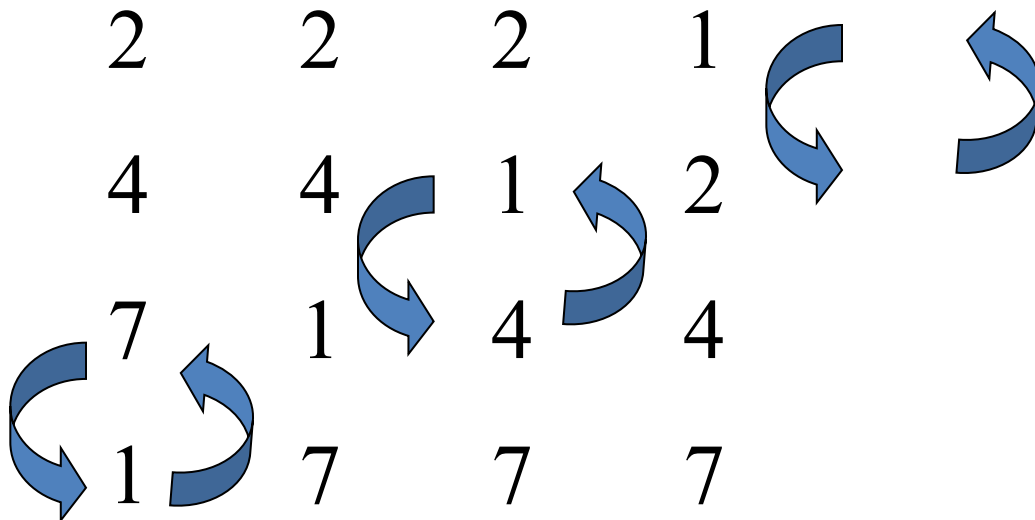
- Нарисуйте в конспекте таблицу

№ вопр.	Ответ	Проверка
3.1		
3.2		
3.3		
3.4		
3.5		

- В качестве ответов используйте:  
I – да, - нет, O – не известно

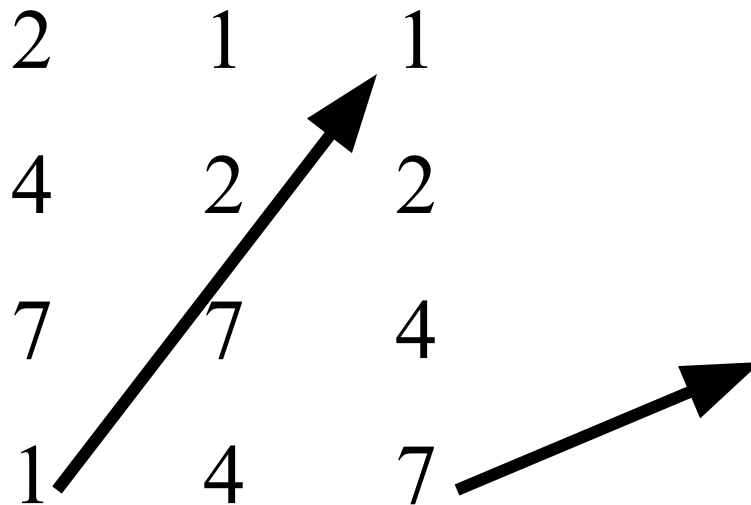
# Вопрос 3.1

- Верно ли, что представленная ниже сортировка чисел реализует метод «пузырька»?



## Вопрос 3.2

- Верно ли, что представленная ниже сортировка чисел реализует метод «вставки»?





## Вопрос 3.3

- Верно ли, что время сортировки методом «выбора» не зависит от конкретных значений данных в сортируемом наборе?

Например:

<i>Массив А</i>	1	2	3	6	4	5
<i>Массив В</i>	6	5	4	3	2	1

## Вопрос 3.4

- Верно ли, что для сортировки последовательных файлов используют метод «слияния»?

## Вопрос 3.5

- Верно ли, что метод «шейкер» сортировки является развитием метода «пузырька»?

## Викторина 3 (ответы)

- Сравните свои ответы и отметьте правильные в графе проверка галочкой

№ вопр.	Ответ	Проверка
3.1	I	
3.2	-	
3.3	-	
3.4	I	
3.5	I	

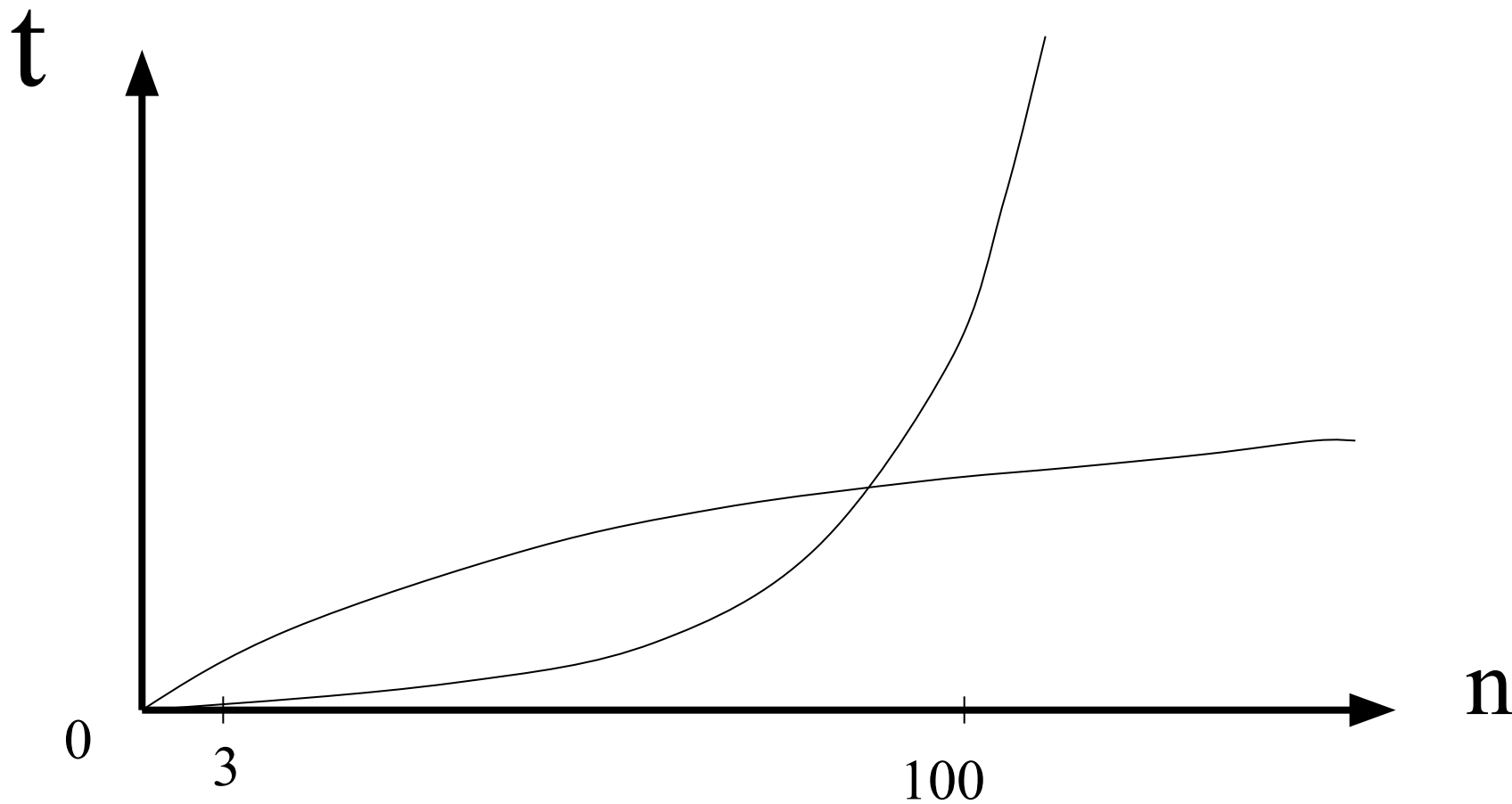
- Количество правильных ответов запишите!

# Анализ эффективности алгоритмов

- Зависимость времени выполнения от объема входных данных
- Зависимость требуемого объема оперативной памяти от объема входных данных

Тета классы  $\Theta(n^2)$  и  $\Theta(\lg n)$

# Пример зависимости времени выполнения алгоритма от числа элементов



# Quicksort

- Quicksort is a well-known sorting algorithm developed by C. A. R. Hoare
- Typically, quicksort is significantly faster in practice than other  $\Theta(n \log n)$  algorithms
- Quicksort is a comparison sort and, in efficient implementations, is not a **stable sort**.



# Sir Charles Antony Richard Hoare

- British computer scientist, probably best known for the development of **Quicksort**, the world's most widely used sorting algorithm, in 1960. He also developed **Hoare logic**.
- Born in **Colombo** (Sri Lanka). he received his Bachelor's degree in Classics from the **University of Oxford** (Merton College) in 1956. he studied computer translation of human languages at **Moscow State University** in the Soviet Union in the school of **Kolmogorov** . As a Professor of Computing to lead the **Programming Research Group** in the **Oxford University Computing Laboratory**.



# Stability

- **Stable sorting algorithms** maintain the relative order of records with equal keys (*i.e.* values). That is, a sorting algorithm is *stable* if whenever there are two records  $R$  and  $S$  with the same key and with  $R$  appearing before  $S$  in the original list,  $R$  will appear before  $S$  in the sorted list.

# Теория решения задач

**Фазы решения задач Дж. Пойя (*Дьердь Пойа* или *Georgt Polia* – 1945 год):**

- Фаза 1. Понять существо задачи
- Фаза 2. Разработать план решения задачи
- Фаза 3. Выполнить план
- Фаза 4. Оценить точность решения, а также его потенциал в качестве средства решения других задач

# Теория Решения Изобретательских Задач (ТРИЗ)

- **Генрих Саулович Альтшуллер** – автор
  - а) вместо действия, диктуемого условиями задачи, осуществить обратное действие;
  - б) сделать движущуюся часть объекта или внешней среды неподвижной, а неподвижную - движущейся;
  - в) повернуть объект "вверх ногами", вывернуть его.

Приемов было выявлено более сорока.

# Литература

- Дж. Гленн Брукшир. Введение в компьютерные науки, «Вильямс», 2001. с. 213-276.
- Niklaus Wirth. Algorithms + Data Structures = Programs. “Prentice-Hall”, 1976. – 366 p.
- D. E. Knuth, [\*The Art of Computer Programming, Volume 3: Sorting and Searching\*](#), 1973.

# Спасибо за внимание!

Dr. Sc Ing. Борис Мишнев

