

---

# Введение

Алгоритмы и структуры данных

Лекция

---

Процесс создания компьютерной программы для решения какой-то практической задачи состоит из:

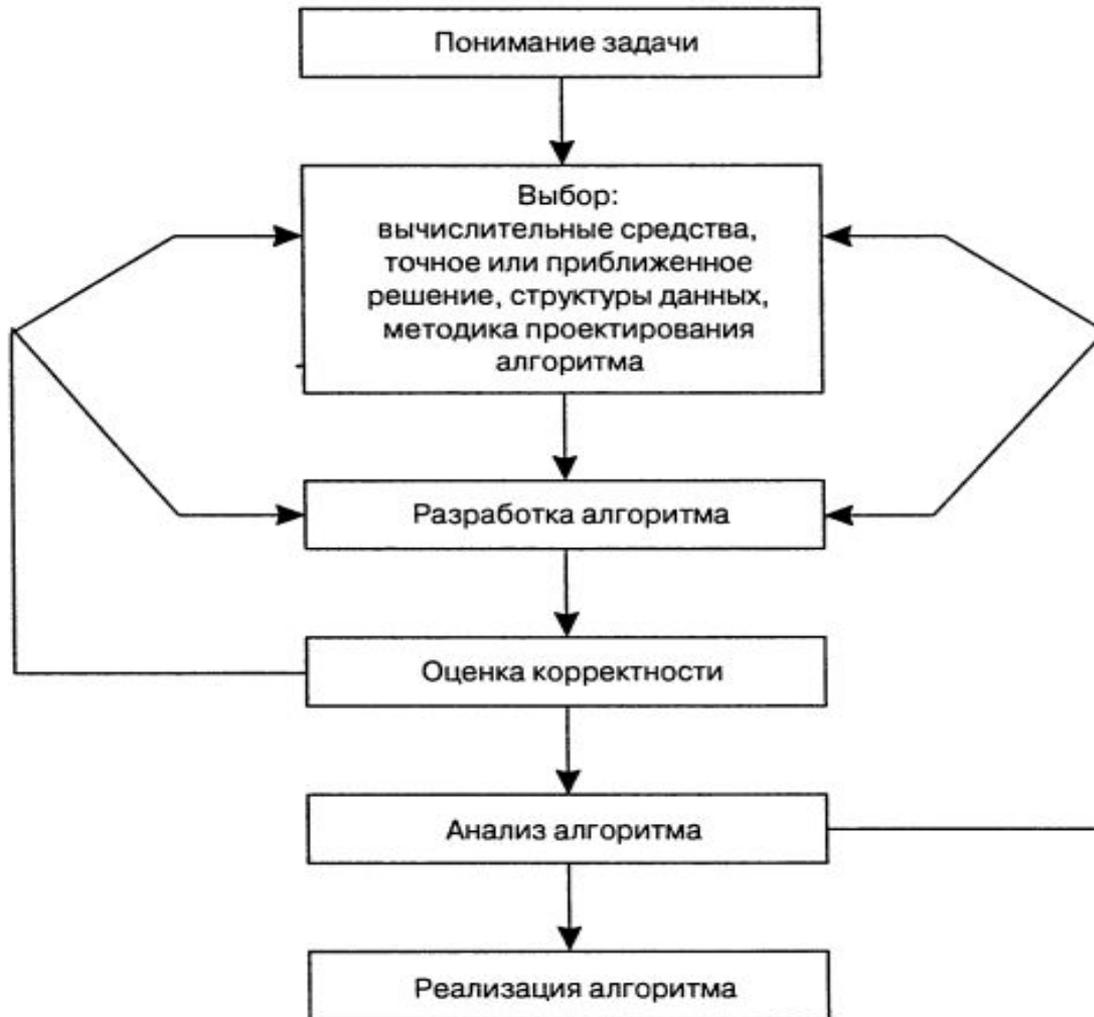
---

- Формализация и создание технического задания на исходную задачу
- Разработка алгоритма решения задачи;
- Написание, тестирование, отладка и документирование программы;
- Получение решения исходной задачи путём выполнения законченной программы



# Разработка алгоритма решения задачи

---



---

**Алгоритм (algorithm)** – это формально описанная вычислительная процедура, получающая **исходные данные** (input), называемые также входом алгоритма или его аргументом, и выдающая **результат вычислений на выход** (output).

---





---

**Алгоритм** считают правильным (correct), если на любом допустимом (для данной задачи) входе он заканчивает и выдает результат, удовлетворяющий требованиям задачи. В этом случае говорят, что алгоритм **решает** (solves) данную вычислительную задачу.

---



# Сложность алгоритма

---

- Временная и емкостная
- В наилучшем и в среднем



# Задача сортировки

---

□ Вход: Последовательность  $N$  чисел  $(a_1, a_2, \dots, a_n)$

Выход: Перестановка  $(a'_1, a'_2, \dots, a'_n)$  исходной последовательности, для которой

$$a'_1 \leq a'_2 \leq \dots \leq a'_n$$



# Сортировка вставками

---

- Входные данные: массив  $A[1..N]$
- Требования: последовательность сортируется «на месте», без дополнительной памяти (помимо массива используется фиксированное число ячеек памяти)
- Результат: массив  $A$  упорядочен по возрастанию



# Сортировка вставками (псевдокод)

---

INSERTION-SORT( $A$ )

```
1  for  $j \leftarrow 2$  to  $\text{length}[A]$ 
2      do  $\text{key} \leftarrow A[j]$ 
3           $\triangleright$  добавить  $A[j]$  к отсортированной части  $A[1..j-1]$ .
4           $i \leftarrow j - 1$ 
5          while  $i > 0$  and  $A[i] > \text{key}$ 
6              do  $A[i+1] \leftarrow A[i]$ 
7                   $i \leftarrow i - 1$ 
8           $A[i+1] \leftarrow \text{key}$ 
```



# Анализ алгоритма вставками

---

Анализируемые параметры:

- Размер входа
- Время работы (это число элементарных шагов, которые он выполняет)



# Анализ алгоритма вставками

INSERTION-SORT( $A$ )	стоимость	число раз
1 <b>for</b> $j \leftarrow 2$ <b>to</b> $length[A]$	$c_1$	$n$
2 <b>do</b> $key \leftarrow A[j]$	$c_2$	$n - 1$
3 $\triangleright$ добавить $A[j]$ к отсортиро-		
$\triangleright$ ванной части $A[1..j - 1]$ .	0	$n - 1$
4 $i \leftarrow j - 1$	$c_4$	$n - 1$
5 <b>while</b> $i > 0$ and $A[i] > key$	$c_5$	$\sum_{j=2}^n t_j$
6 <b>do</b> $A[i + 1] \leftarrow A[i]$	$c_6$	$\sum_{j=2}^n (t_j - 1)$
7 $i \leftarrow i - 1$	$c_7$	$\sum_{j=2}^n (t_j - 1)$
8 $A[i + 1] \leftarrow key$	$c_8$	$n - 1$



# Анализ алгоритма вставками (время работы)

---

$$T(n) = c_1 n + c_2(n - 1) + c_4(n - 1) + c_5 \sum_{j=2}^n t_j + \\ + c_6 \sum_{j=2}^n (t_j - 1) + c_7 \sum_{j=2}^n (t_j - 1) + c_8(n - 1).$$



# Анализ алгоритма вставками (время работы)

---

**Наилучший случай – упорядоченная последовательность**

$$\begin{aligned}T(n) &= c_1n + c_2(n - 1) + c_4(n - 1) + c_5(n - 1) + c_8(n - 1) = \\ &= (c_1 + c_2 + c_4 + c_5 + c_8)n - (c_2 + c_4 + c_5 + c_8).\end{aligned}$$

$$T(n) = an + b$$

---



# Анализ алгоритма вставками (время работы)

---

**Наихудший случай – массив расположен в обратном порядке**

$$\begin{aligned} T(n) &= c_1 n + c_2(n-1) + c_4(n-1) + c_5 \left( \frac{n(n+1)}{2} - 1 \right) + \\ &\quad + c_6 \left( \frac{n(n-1)}{2} \right) + c_7 \left( \frac{n(n-1)}{2} \right) + c_8(n-1) = \\ &= \left( \frac{c_5}{2} + \frac{c_6}{2} + \frac{c_7}{2} \right) n^2 + \left( c_1 + c_2 + c_4 + \frac{c_5}{2} - \frac{c_6}{2} - \frac{c_7}{2} + c_8 \right) n - \\ &\quad - (c_2 + c_4 + c_5 + c_8). \end{aligned}$$

$$T(n) = an^2 + bn + c.$$

---



# Важные типы задач

---

- Сортировка
- Поиск
- Обработка строк
- Задачи из теории графов
- Комбинаторные задачи
- Геометрические задачи
- Численные задачи

