

Основная литература



- Ахо, А.В. Структуры данных и алгоритмы / А.В. Ахо, Дж. Хопкрофт, Дж.Д. Ульман. – М.: Издательский дом «Вильямс», 2001. – 384 с.
- Вирт, Н. Алгоритмы и структуры данных = Algorithms and data structures / Н. Вирт. – СПб.: Невский проспект, 2001. – 351 с.
- Ахо, А. Построение и анализ вычислительных алгоритмов / А. Ахо, Дж. Хопкрофт, Дж. Ульман. – М.: Издательство «Мир», 1979. – 519 с.
- Кормен, Томас Х., Лейзерсон, Чарльз И., Ривест, Рональд Л., Штайн, Клиффорд. Алгоритмы: построение и анализ, 2-е издание. : Пер. с англ. — М. : Издательский дом "Вильямс", 2005. — 1296 с.
- Дональд Э. Кнут, Искусство программирования, Т. 1, Т. 3.

Дополнительная литература



- Х. М. Дейтел, П. Дж. Дейтел - Как программировать на С, М. : Бином-пресс, 2006. — 912 с.

Лекция 1



А.Ф. ЗУБАИРОВ

Понятие алгоритма



- Алгоритм - точный набор инструкций, описывающих порядок действий исполнителя для достижения результата решения задачи за конечное время.
- Алгоритм – набор конечного числа правил, задающих последовательность выполнения операций для решения задач определенного типа (Кнут).
- Алгоритм – формально описанная вычислительная процедура, получающая **входные данные**, называемые входом алгоритма или его аргументом, и выдающая результат вычисления на выход (Кормен).
- Алгоритм – конечная последовательность инструкций, каждая из которых имеет чёткий смысл и может быть выполнена с конечными вычислительными затратами за конечное время (Ахо).

Алгоритм Евклида



● **Алгоритм А** (Алгоритм Евклида). Даны два целых положительных числа m и n . Требуется найти их наибольший общий делитель.

А1. [Нахождение остатка]. Разделим m на n , пусть остаток от деления будет равен r (где $0 \leq r < n$).

А2. [Сравнение с нулём]. Если $r = 0$, то выполнение алгоритма прекращается; n – искомое значение.

А3. [Замещение] Присвоить $m \leftarrow n, n \leftarrow r$ и вернуться к шагу А1.

Свойства алгоритма



- Конечность - при корректно заданных исходных данных алгоритм должен заканчиваться после выполнения конечного числа шагов.
- Определенность (детерминированность) - в каждый момент времени каждый шаг алгоритма должен быть точно определён. Таким образом, алгоритм выдаёт один и тот же результат (ответ) для одних и тех же исходных данных.
- Понятность — алгоритм для исполнителя должен включать только те команды, которые входят в его систему команд.
- Завершаемость (конечность) — при корректно заданных исходных данных алгоритм должен завершать работу и выдавать результат за конечное число шагов.
- Эффективность (Кнут) — алгоритм считается эффективным, если все его операторы достаточно просты для того, чтобы их можно было выполнить в течение конечного промежутка времени с помощью карандаша и бумаги.
- Массовость (универсальность). Алгоритм должен быть применим к разным наборам исходных данных.

Ввод и вывод алгоритма



- Ввод: алгоритм имеет некоторое (возможно, равное 0) число входных данных, т.е. величин, которые задаются до начала его работы или определяются динамически во время его работы.
- Вывод: алгоритм имеет одно или несколько выходных данных, т.е. величин, имеющих вполне определённую связь с входными данными.

Алгоритмы и их сложности



- *Размер задачи* – некое число, выражающее меру количества входных данных.
- *Временная сложность* – время, затрачиваемое алгоритмом как функция размера задачи.
- *Асимптотическая временная сложность* – поведение сложности в пределе при увеличении размера задачи.
- Аналогично можно определить емкостную сложность и асимптотическую емкостную сложность.

Асимптотическая сложность



- Если алгоритм обрабатывает входы размера n за время cn^2 , где c – const, говорят, что временная сложность алгоритма есть $O(n^2)$ (читается порядка n^2).
- Неотрицательная функция $g(n)$ есть $O(f(n))$, если существует такая постоянная c , что $g(n) \leq cf(n)$ для всех, кроме конечного множества, неотрицательных значений n .

Эффективные алгоритмы



● $A_1 - A_5$ – алгоритмы.

Алгоритм	Временная сложность	Максимальный размер задачи		
		1 с	1 мин	1 час
A_1		1000	$6 \cdot 10^4$	$3,6 \cdot 10^6$
A_2		140	4893	$2,0 \cdot 10^5$
A_3		31	244	1897
A_4		10	39	153
A_5		9	15	21

Тип данных



- Тип данных – формы представления данных, которые изначально существуют и могут обрабатываться в языке программирования.
- Все формы представления данных в конкретном языке, имеющиеся изначально, называются *базовыми типами данных*.

Концепция типа данных



- Каждая константа, переменная, выражение или функция имеют определенный тип.
- 1. Любой тип определяет множество значений, к которым относится некоторая константа, которое может принимать некоторая переменная или выражение и которое может формироваться операцией или функцией.
- 2. Тип любой величины, обозначаемой константой, переменной или выражением, может быть выведен по её виду или описанию.
- 3. Каждая операция или функция требует аргументы определенного типа и дает результаты также фиксированного типа.

Производные (новые) типы данных



- Новые типы данных определяются с помощью ранее определенных типов данных.
- Значения, принадлежащие такому типу обычно представляют собой совокупности *значений компонент*, принадлежащие к определенным *типам компонент*; такие составные значения называются структурированными.

Кардинальное число типа



- Кардинальное число T – число различных значений, принадлежащих типу T .
- Кардинальное число определяет размер памяти, нужной для размещения переменной x типа T . Обозначается $x:T$.

Структура данных



- Совокупности данных, организованные некоторым образом, называются *структурами данных*.
- *Последовательные структуры* – структура данных хранится в едином непрерывно блоке памяти.
- *Связанные структуры* – о взаимном расположении элементов в памяти компьютера заранее неизвестно.
- *Линейные структуры* – такие, в которых однозначно определен порядок следования элементов.
- *Нелинейные структуры* – такие, в каком порядок следования элементов может быть любым.