

Основы организации данных и алгоритмизация (2012)

Лекция 1. Алгоритмы и их свойства

Штанюк А.А.

9 декабря 2012 г.



- 1 Введение
- 2 Понятие алгоритма
- 3 Способы записи
- 4 Классы алгоритмов
- 5 Классификация алгоритмов
- 6 Вопросы для самоконтроля
- 7 Глоссарий
- 8 Библиографический список

- 1 Введение
- 2 Понятие алгоритма
- 3 Способы записи
- 4 Классы алгоритмов
- 5 Классификация алгоритмов
- 6 Вопросы для самоконтроля
- 7 Глоссарий
- 8 Библиографический список

Введение

ПРОГРАММА

Компьютерная **программа** - центральное понятие, объект исследования и разработки. Она представляет собой запись на формализованном языке алгоритма решения задачи. Решение задачи заключается в последовательном применении к исходным данным некоторых операций до тех пор, пока не будет получено то, что называется результатом.

Возникают следующие вопросы:

- 1 Что такое алгоритм?
- 2 Как организовать данные в памяти ЭВМ?
- 3 Из каких составных элементов строится программа?
- 4 Как доказать правильность работы программы?
- 5 Как оценить эффективность алгоритма?
- 6 Как правильно выбрать алгоритм для решения поставленной задачи?

- 1 Введение
- 2 Понятие алгоритма**
- 3 Способы записи
- 4 Классы алгоритмов
- 5 Классификация алгоритмов
- 6 Вопросы для самоконтроля
- 7 Глоссарий
- 8 Библиографический список

Понятие алгоритма

Определение алгоритма

Алгоритм - предписание, однозначно задающее процесс преобразования исходной информации в виде последовательности элементарных дискретных шагов, приводящих за их конечное число к результату.

Слово **алгоритм** арабского происхождения и связано с именем математика и астронома *Абу Абдулла Абу Джафар Мухамад ибн Муса ал-Хорезми*, который в 825 г. н.э. описал придуманную в Индии позиционную десятичную систему исчисления.



Свойства алгоритмов

Основные свойства алгоритма:

- 1 **Дискретность.** Процесс решения протекает в виде последовательности отдельных действий, следующих друг за другом.
- 2 **Элементарность действий.** Каждое действие является настолько простым, что оно не допускает возможности неоднозначного толкования.
- 3 **Определенность.** Каждое действие определено и после выполнения каждого действия однозначно определяется, какое действие будет выполнено следующим.
- 4 **Связанность.** На каждом следующем шаге используются результаты предыдущих.
- 5 **Конечность.** Алгоритм заканчивает работу после конечного числа шагов.
- 6 **Результативность.** В момент прекращения работы алгоритма известно, что является результатом.
- 7 **Массовость.** Алгоритм описывает некоторое множество процессов, применимых при различных входных данных.

Понятие алгоритма

Алгоритм считается **правильным**, если при любых допустимых данных он заканчивает работу и выдает результат, удовлетворяющий требованиям задачи.

Алгоритм **однозначен**, если при применении к одним и тем же входным данным он дает один и тот же результат.

Для работы алгоритма требуется **исполнитель** - человек или механизм, понимающий способ записи и интерпретации действий.

Алгоритмизация

Алгоритмизация - процесс систематического составления алгоритмов для решения поставленных задач. Она является необходимым шагом в процессе разработки программного обеспечения.

Для составления алгоритмов нужны творческие способности, знание теории и способов записи алгоритма. Для исполнения алгоритма необходимо неукоснительно следовать каждому шагу.

Теоретическое определение алгоритма

Определение *А.Н. Колмогорова*:

Алгоритмом принято называть систему вычислений, которая для некоторого класса математических задач из записи A «условий» задачи позволяет при помощи однозначно определенной последовательности операций, совершаемых «механически», без вмешательства творческих способностей человека, получить запись B «решения» задачи.

Таким образом, алгоритм - это процедура, которая:

- 1 Применяется к строго определенным исходным данным A .
- 2 Представляется в виде последовательности простых шагов; каждый шаг состоит в непосредственной обработке возникшего к этому шагу состояния S в состояние $S^* = \Omega(S)$.
- 3 Преобразование S в S^* производится однозначно заданным способом.
- 4 Процесс переработки A продолжается до тех пор, пока либо не произойдет безрезультатная остановка, либо не появится сигнал о получении решения B . При этом не исключается возможность непрекращающегося процесса переработки.

Теоретическое определение алгоритма

$$A_0 = A$$

$$A_1 = \Omega(A_0)$$

$$A_2 = \Omega(A_1)$$

...

$$A_n = \Omega(A_{n-1})$$

...

$$B = A_k$$

Каждый алгоритм задает функцию, которая по набору исходных данных выдает результат применения алгоритма к этим данным. Такая функция называется *вычислимой*.

- | | |
|-------------------------|----------------------------|
| 1 Введение | 5 Классификация алгоритмов |
| 2 Понятие алгоритма | 6 Вопросы для самоконтроля |
| 3 Способы записи | 7 Глоссарий |
| 4 Классы алгоритмов | 8 Библиографический список |

Способы записи алгоритма

- Словесный (естественный язык, ЕЯ)
- Ограниченный естественный язык (ОЕЯ)
- Схематический (графический)
- Псевдоязык (в том числе языки программирования)

Использование естественного языка

Наиболее простой способ - ЕЯ (естественный язык)

Пример алгоритма для приготовления кипятка

- 1 взять чайник
- 2 открыть крышку
- 3 если есть вода, перейти к шагу 5
- 4 налить воды
- 5 поставить на огонь
- 6 дожидаться струи пара

Использование естественного языка

Достоинства:

- 1 Понятен всем носителям языка
- 2 Не требует специального синтаксиса
- 3 Легко поддаётся редактированию

Недостатки:

- 1 Неоднозначен (некоторые слова и выражения могут толковаться по-разному)
- 2 Трудно изучаем для иностранцев, не владеющих языком
- 3 Несёт в себе избыточность в плане количества информации

Использование ограниченного естественного языка

Мы можем устранить неоднозначность, оставив только *однозначные* слова.

Пример

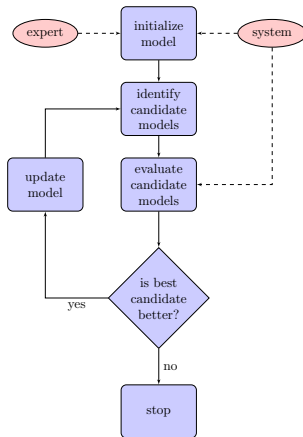
- 1 Посмотреть налево
- 2 Посмотреть направо
- 3 Перейти дорогу

Каждое слово может быть *командой* или *параметром*. В примере слова **Перейти**, **Посмотреть** относятся к командам, а остальные - к параметрам.

Хороший пример командного языка: **армейский**. В условиях боевых действий нет времени задумываться о смысле заданий.

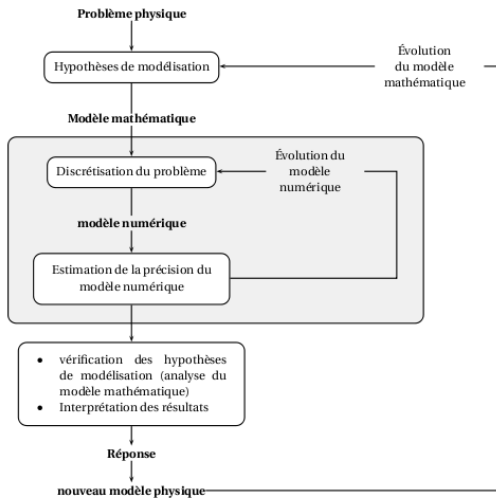
Использование схем и диаграмм

Схемы, рисунки и диаграммы часто бывают более наглядны, но могут требовать и пояснений.



Использование схем и диаграмм

В данной схеме сочетаются языки **блок-схем (flowchart)** и французский.



Использование схем и диаграмм

Элементы блок-схем строятся (согласно **ГОСТ 19.701-90**) из следующих элементов:

Использование схем и диаграмм

Синтаксические диаграммы:



Использование псевдоязыка

Псевдоязык очень напоминает алгоритмический язык программирования, но использует обобщённые для многих языков конструкции

Алгоритм Евклида

```
1: procedure EUCLID( $a, b$ )                                ▷ НОД  $a$  и  $b$ 
2:    $r \leftarrow a \bmod b$ 
3:   while  $r \neq 0$  do                                    ▷ ответ получен, если  $r$  равно 0
4:      $a \leftarrow b$ 
5:      $b \leftarrow r$ 
6:      $r \leftarrow a \bmod b$ 
7:   end while
8:   return  $b$                                              ▷ НОД -  $b$ 
9: end procedure
```

Использование псевдоязыка

Блок IF

```
1: if  $quality \geq 9$  then
2:    $a \leftarrow perfect$ 
3: else if  $quality \geq 7$  then
4:    $a \leftarrow good$ 
5: else if  $quality \geq 5$  then
6:    $a \leftarrow medium$ 
7: else if  $quality \geq 3$  then
8:    $a \leftarrow bad$ 
9: else
10:    $a \leftarrow unusable$ 
11: end if
```

Блок WHILE

```
1:  $sum \leftarrow 0$ 
2:  $i \leftarrow 1$ 
3: while  $i \leq n$  do
4:    $sum \leftarrow sum + i$ 
5:    $i \leftarrow i + 1$ 
6: end while
```

Блок REPEAT

```
1:  $sum \leftarrow 0$ 
2:  $i \leftarrow 1$ 
3: repeat
4:    $sum \leftarrow sum + i$ 
5:    $i \leftarrow i + 1$ 
6: until  $i > n$ 
```

Использование псевдоязыка

Блок **FOR**

```
1:  $sum \leftarrow 0$   
2: for  $i \leftarrow 1, n$  do  
3:    $sum \leftarrow sum + i$   
4: end for
```

Пример:

Небольшой псевдокод...

```
1:  $s \leftarrow 0$   
2:  $p \leftarrow 0$   
3: for  $i \leftarrow 1, 10$  do  
4:    $s \leftarrow s + i$   
5:    $p \leftarrow p + s$   
6: end for
```

- | | |
|----------------------------|----------------------------|
| 1 Введение | 5 Классификация алгоритмов |
| 2 Понятие алгоритма | 6 Вопросы для самоконтроля |
| 3 Способы записи | 7 Глоссарий |
| 4 Классы алгоритмов | 8 Библиографический список |

Схема определения алгоритма на ЭВМ

- 1 Всякий алгоритм применяется к исходным данным и выдает результирующие данные. В ходе работы алгоритма появляются промежуточные данные. Для описания данных фиксируется набор элементарных символов (**алфавит** данных) и даются правила построения сложных данных из простых. Примеры простых данных: целые и действительные числа, логические переменные, символьные переменные. Примеры сложных данных: массивы, строки, структуры.
- 2 Данные для своего размещения требуют **памяти**. В ЭВМ память состоит из ячеек. Единицы объема данных и памяти согласованы, и в прикладных алгоритмических моделях объем данных можно измерять числом ячеек, в которых данные размещены.
- 3 Элементарные шаги алгоритма состоят из **базовых действий**, число которых конечно. Под ними можно подразумевать машинные команды, входящие в набор команд ЭВМ. При записи алгоритмов на языках высокого уровня в качестве базовых действий могут выступать операторы языка.

Классы алгоритмов

Можно выделить три крупных класса алгоритмов: *вычислительные, информационные и управляющие.*

Вычислительные алгоритмы , как правило, работают с простыми видами данных (числа, матрицы), но сам процесс вычисления может быть долгим и сложным.

Информационные алгоритмы представляют собой набор сравнительно небольших процедур (например, поиск числа, слова), но работающих с большими объемами информации (базы данных). Для того чтобы они работали эффективно, важно иметь хорошую организацию данных.

Управляющие алгоритмы характерны тем, что данные к ним поступают от внешних процессов, которыми они управляют. Результатом работы этих алгоритмов являются различные управляющие воздействия.

- | | | | |
|----------|-------------------|----------|---------------------------------|
| 1 | Введение | 5 | Классификация алгоритмов |
| 2 | Понятие алгоритма | 6 | Вопросы для самоконтроля |
| 3 | Способы записи | 7 | Глоссарий |
| 4 | Классы алгоритмов | 8 | Библиографический список |

Классификация алгоритмов

Алгоритмы можно классифицировать по множеству признаков:

- ▶ по функциональному назначению
- ▶ по временной и пространственной сложности
- ▶ по сложности реализации
- ▶ по эффективности исполнения
- ▶ по точности при решении приближённых задач
- ▶ и др.

Рассмотрим некоторые категории, к которым можно отнести известные вычислительные алгоритмы

Классификация алгоритмов

- ▶ **Общие комбинаторные алгоритмы** - связаны с генерацией комбинаций элементов
- ▶ **Алгоритмы на графах** - всевозможные операции с использованием структуры данных **граф**
- ▶ **Алгоритмы на строках** - связаны с решением задач, относящихся к строкам и символам
- ▶ **Алгоритмы сортировки** - упорядочивание элементов
- ▶ **Алгоритмы сжатия данных** - преобразование данных с устранением избыточности информации

Классификация алгоритмов

- ▶ **Вычислительная геометрия** - всевозможные геометрические задачи
- ▶ **Компьютерная графика** - рисование геометрических фигур на экране ЭВМ
- ▶ **Криптографические алгоритмы** - шифрование/дешифрование данных
- ▶ **Цифровая обработка сигналов** - преобразование данных методами ЦОС
- ▶ **Алгоритмы операционных систем** - выделение и освобождение памяти в операционной системе, планирование процессов, устранение взаимоблокировок и др.
- ▶ **Сетевые алгоритмы** - расчёт маршрутов, потоков в сетях
- ▶ **Генетические алгоритмы** - использование генетического механизма для обработки компьютерных данных

Классификация алгоритмов

- ▶ **Алгоритмы нейронных сетей** - использование принципов обработки информацией мозгом в ЭВМ
- ▶ **Вычислительная алгебра** - решение широкого спектра алгебраических задач
- ▶ **Теоретико-числовые алгоритмы** - операции над числами
- ▶ **Численные алгоритмы** - решение задач приближёнными методами
- ▶ **Алгоритмы оптимизации** - нахождение оптимальных значений параметров систем
- ▶ **Алгоритмы синтаксического и грамматического разбора** - используются для обработки текстов на формальных и естественных языках
- ▶ **Другие...**

- 1 Введение
- 2 Понятие алгоритма
- 3 Способы записи
- 4 Классы алгоритмов
- 5 Классификация алгоритмов
- 6 Вопросы для самоконтроля**
- 7 Глоссарий
- 8 Библиографический список

Вопросы для самоконтроля

- Какую роль играет понятие алгоритма в программировании?
▶ Подсказка
- Что такое алгоритм? Откуда произошло понятие алгоритма?
▶ Подсказка
- Что мы относим к свойствам алгоритма? ▶ Подсказка
- Какие свойства алгоритма относятся к важнейшим? ▶ Подсказка
- Что такое исполнитель алгоритма? ▶ Подсказка
- Какой алгоритм считается правильным? Однозначным? ▶ Подсказка
- Что такое алгоритмизация? ▶ Подсказка
- Какие способности нужны разработчику алгоритма? Исполнителю?
▶ Подсказка
- Чем отличается теоретическое определение алгоритма от общепринятого? ▶ Подсказка
- До каких пор будет выполняться алгоритм? ▶ Подсказка

Вопросы для самоконтроля

- Что такое вычислимая функция? Как она связана с алгоритмом?
▶ Подсказка
- Перечислите способы записи алгоритма. ▶ Подсказка
- Приведите пример алгоритма, записанного на естественном языке.
▶ Подсказка
- Перечислите достоинства и недостатки ЕЯ для записи алгоритмов.
▶ Подсказка
- Как от ЕЯ переходят к ОЕЯ? ▶ Подсказка
- В чём преимущества графических схем и рисунков? ▶ Подсказка
- Из каких элементов строятся блок-схемы? ▶ Подсказка
- Что такое псевдоязык? ▶ Подсказка
- В чём заключается схема определения алгоритма на ЭВМ? ▶ Подсказка
- Как согласуются между собой данные и память? Машинные команды и шаги алгоритма? ▶ Подсказка

Вопросы для самоконтроля

- Перечислите три крупных класса алгоритмов. В чём их особенности?

► Подсказка

- Как можно классифицировать алгоритмы?

► Подсказка

- 1 Введение
- 2 Понятие алгоритма
- 3 Способы записи
- 4 Классы алгоритмов
- 5 Классификация алгоритмов
- 6 Вопросы для самоконтроля
- 7 Глоссарий
- 8 Библиографический список

Глоссарий I

Алгоритм конечная последовательность дискретных шагов, приводящая к решению задачи. 6

Алгоритмизация процесс составления алгоритмов для решения поставленных задач. 9

Блок-схема графическое представление алгоритма. 13, 17, 19

ЕЯ естественный язык. Язык, используемый для общения в самом широком смысле. 13

ОЕЯ ограниченный естественный язык. Язык, из которого исключены неоднозначные слова. 13

Псевдоязык язык, используемый для записи алгоритмов в форме, близкой к языкам программирования. 13

- 1 Введение
- 2 Понятие алгоритма
- 3 Способы записи
- 4 Классы алгоритмов
- 5 Классификация алгоритмов
- 6 Вопросы для самоконтроля
- 7 Глоссарий
- 8 Библиографический список

Библиографический список I



Кормен Т., Лейзерсон Ч., Ривест Р.

Алгоритмы: построение и анализ

МЦНМО, Москва, 2000



Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ.

2-е изд. — М.: «Вильямс», 2006



Википедия

<http://ru.wikipedia.org/wiki/Алгоритм>



Википедия

http://ru.wikipedia.org/wiki/Список_алгоритмов



Серджвик Р.

Фундаментальные алгоритмы на C++. Части 1-4

Diasoft, 2001

Библиографический список II



Седжвик Р.

Фундаментальные алгоритмы на С. Анализ/Структуры данных/Сортировка/Поиск

СПб.: ДиаСофтЮП, 2003



Седжвик Р.

Фундаментальные алгоритмы на С. Алгоритмы на графах

СПб.: ДиаСофтЮП, 2003



Ахо А., Хопкрофт Д., Ульман Д. Структуры данных и алгоритмы.

Издательский дом «Вильямс», 2000



Кнут Д.

Искусство программирования, том 1. Основные алгоритмы

3-е изд. — М.: «Вильямс», 2006



Кнут Д.

Искусство программирования, том 2. Получисленные методы

3-е изд. — М.: «Вильямс», 2007

Библиографический список III



Кнут Д.

Искусство программирования, том 3. Сортировка и поиск
2-е изд. — М.: «Вильямс», 2007



Кнут Д.

Искусство программирования, том 4, выпуск 3. Генерация всех сочетаний
и разбиений
М.: «Вильямс», 2007



Кнут Д.

Искусство программирования, том 4, выпуск 4. Генерация всех деревьев.
История комбинаторной генерации
М.: «Вильямс», 2007



Керниган Б., Ритчи Д.

Язык программирования Си
М.: Финансы и статистика,



Подбельский В.В., Фомин С.С.

Программирование на языке С.
М.: Финансы и статистика, 2003

Библиографический список IV

-  *Подбельский В.В.*
Практикум по программированию на языке Си.
М.: Финансы и статистика, 2004
-  *Шилдт Г.*
Полный справочник по С.
М.: Вильямс, 2002.
-  *Дейтел Х., Дейтел П.*
Как программировать на С.