

Московский государственный технический университет
имени Н.Э. Баумана

П.Г. Ключарев, Д.А. Жуков

ВВЕДЕНИЕ В ТЕОРИЮ АЛГОРИТМОВ

Рекомендовано
Научно-методическим советом МГТУ им. Н.Э. Баумана
в качестве учебного пособия

Москва
Издательство МГТУ им. Н.Э. Баумана
2012

УДК 510.5(075.8)

ББК 22.12

К52

Рецензенты: *А.Н. Велигура, А.Ю. Голубков*

Ключарев П. Г.

К52 Введение в теорию алгоритмов : учеб. пособие / П.Г. Ключарев, Д.А. Жуков. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2012. – 37, [3] с.: ил.

Рассмотрены машины Тьюринга, вопросы алгоритмической разрешимости, основные классы сложности, NP -полнота, схемная сложность.

Для студентов МГТУ им. Н.Э. Баумана, обучающихся по специальностям «Информационная безопасность автоматизированных систем» и «Компьютерная безопасность». Пособие может быть полезно студентам других специальностей, связанных с информатикой, вычислительной техникой и информационной безопасностью.

УДК 510.5(075.8)

ББК 22.12

ВВЕДЕНИЕ

В любой области науки и техники и просто в повседневной жизни человек постоянно сталкивается с решением различных задач. Примерами задач могут быть подсчет стоимости продуктов для банкета, выбор оптимального маршрута для поездки в другой конец города, вычисление квадратного корня из заданного числа, расчет оптимальной траектории космического аппарата, шифрование файла.

Решая любую конкретную задачу, человек выполняет некоторую последовательность действий. Формальное описание такой последовательности действий называют алгоритмом. Несмотря на то что алгоритмы могут быть предназначены для решения совершенно разных задач и быть совсем непохожими друг на друга, существуют свойства, характерные для всех алгоритмов.

Общие свойства алгоритмов изучает теория алгоритмов, которой и посвящено настоящее учебное пособие. Теория алгоритмов представляет собой область дискретной математики, находящуюся на стыке математической логики и информатики. Изучение общих свойств задач и алгоритмов с математической точки зрения имеет огромное значение как для чистой математики, так и для большинства областей информатики, прежде всего для программирования и информационной безопасности.

Одним из краеугольных камней информационной безопасности является криптография. Многие методы криптографии основаны на тех или иных свойствах алгоритмов. В частности, почти все современные криптографические алгоритмы и протоколы базируются на односторонних функциях, т. е. на функциях, которые можно быстро вычислить, но для обращения которых требуется очень много времени.

В учебном пособии приведены основные понятия теории алгоритмов. Эта теория непрерывно развивается и накопила знания, далеко выходящие за рамки этого пособия. Заинтересованный читатель найдет много дополнительных сведений в книгах, приведенных в списке литературы [1–16].

1. ОСНОВНЫЕ ПОНЯТИЯ

Учебное пособие рассчитано на читателя, который освоил теорию булевых функций, теорию графов и теорию вероятностей. Читателю, не изучавшему эти разделы математики, мы рекомендуем ознакомиться с ними, используя, например, работы [6, 11, 15].

Далее мы будем применять некоторые стандартные обозначения, в частности:

$\mathbb{N} = \{1; 2; 3; \dots\}$ – множество натуральных чисел;

$\mathbb{R}$ – множество действительных чисел.

Сначала введем основные термины и определения.

Определение 1. Рассмотрим две функции: $f, g: \mathbb{N} \rightarrow \mathbb{N}$. Формула $f = O(g)$ означает, что существуют такие $c \in \mathbb{R}$ и $n_0 \in \mathbb{N}$, что при любом $n > n_0$ выполняется $f(n) \leq cg(n)$.

Определение 2. Формула $f = \Omega(g)$ означает, что $g = O(f)$.

Определение 3. Формула $f = \Theta(g)$ означает, что $f = O(g)$ и $g = O(f)$.

Определение 4. Назовем функцию $f: \mathbb{N} \rightarrow \mathbb{N}$ полиномиальной, если существует такое $k \in \mathbb{R}$, что $f(n) = O(n^k)$.

Дадим теперь более строгое определение понятия задачи, с тем чтобы это понятие можно было теоретически исследовать.

Определение 5. Назовем *абстрактной задачей* некоторое бинарное отношение между элементами двух множеств: множества условий и множества решений.

Примером является задача об умножении двух чисел: условие – пара чисел, а решение – одно число.

Определение 6. Задачи, у которых множество решений состоит из двух элементов (например, ДА и НЕТ), будем называть *задачами распознавания*.

Примером задачи распознавания может служить следующая задача: определить, является ли данный граф гамильтоновым.

Понятие задачи тесно связано с понятием языка.

Определение 7. *Алфавитом* называется конечное непустое множество символов.

Часто используется алфавит $\{0;1\}$, называемый *двоичным*, или *бинарным*, алфавитом.

Определение 8. *Словом* над алфавитом A называется набор символов из алфавита A .

Определение 9. *Языком* над алфавитом A называется множество слов над алфавитом A .

Каждой задаче распознавания можно поставить в соответствие язык. Слово принадлежит этому языку тогда и только тогда, когда для этого входного слова ответ задачи – ДА. В то же время любому языку можно поставить в соответствие задачу распознавания этого языка. В связи с этим мы будем употреблять термины «язык» и «задача распознавания» как синонимы.

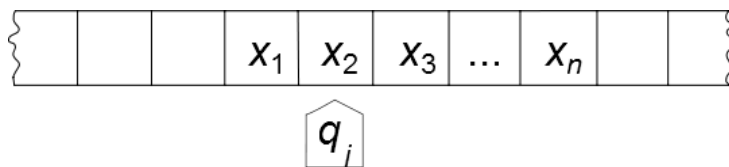
В качестве примера языка можно привести язык над алфавитом $\{0; 1\}$, состоящий из записей всех простых чисел в двоичной системе счисления. Соответствующая задача распознавания формулируется так: является ли данное число простым?

Условие задачи можно некоторым разумным образом закодировать над конечным алфавитом (следовательно, и над двоичным алфавитом). Например, рациональное число всегда можно закодировать над двоичным алфавитом, используя представление с плавающей точкой. Граф можно закодировать с помощью матрицы смежности, матрицы инцидентности или списка вершин. Аналоговый сигнал можно закодировать в виде последовательности отсчетов, взятых с некоторым периодом дискретизации, каждый из которых закодирован в виде двоичного числа. Конкретный способ кодирования следует выбирать, исходя из специфики задачи.

Во введении мы говорили об алгоритме как о некоторой последовательности действий. Такое определение нельзя признать удовлетворительным. Действительно, совершенно непонятно, что такое «действия» и кто их выполняет. Поэтому первой задачей теории алгоритмов является строгое определение понятия алгоритма. Делается это путем постулирования некоторой конструкции, удовлетворяющей интуитивным представлениям об алгоритме. Такая конструкция называется *алгоритмической моделью*. Было предложено большое число различных алгоритмических моделей. С одной из таких моделей – машиной Тьюринга – мы и познакомимся.

Определение 10. *Машина Тьюринга* состоит из *головки* и *ленты*, по которой эта головка перемещается (см. рисунок). Лента

бесконечна в обе стороны и разделена на ячейки, причем каждая ячейка хранит символ из некоторого конечного алфавита A . У машины Тьюринга имеется внутреннее состояние, которое выбирается из некоторого конечного множества Q . Некоторые состояния выделяются особо и называются допускающими. Головка машины Тьюринга в любой момент времени обозревает какую-либо одну клетку ленты.



Машина Тьюринга

Работает машина Тьюринга по шагам. В начальный момент времени на ленте записан *вход*, представляющий собой конечный набор символов, а головка обозревает ячейку, содержащую первый (крайний левый) символ из этого набора. Все остальные ячейки ленты содержат пробелы. На каждом шаге с помощью некоторой заданной *функции переходов* исходя из символа, записанного в обозреваемой головкой ячейке, и текущего состояния выбираются символ, записываемый в обозреваемую клетку, новое состояние и направление перехода головки. Направлений перехода в нашей формализации будет три:

- 1) влево (L) – головка сдвигается на одну клетку влево;
- 2) вправо (R) – головка сдвигается на одну клетку вправо;
- 3) на месте (S) – головка остается там же, где была.

Машина Тьюринга завершает работу в двух случаях:

1) если очередное состояние принадлежит заранее определенному подмножеству множества состояний – множеству допускающих состояний F . В этом случае говорят, что машина Тьюринга остановилась в допускающем состоянии;

2) если функция переходов не определена на аргументе, соответствующем символу обозреваемой клетки и текущему состоянию. В этом случае говорят, что машина Тьюринга остановилась в недопускающем состоянии.

Более формально машиной Тьюринга называется набор

$$MT = (A, Q, F, q_0, \delta), \quad (1)$$

где A – алфавит, причем он обязательно содержит пробельный символ; Q – конечное множество состояний; F – множество допускающих состояний, причем $F \subseteq Q$; q_0 – начальное состояние, $q_0 \in Q$; δ – частично определенная функция переходов машины Тьюринга:

$$\delta: A \times Q \rightarrow A \times Q \times \{L, S, R\}. \quad (2)$$

Функцию δ удобно задавать в виде таблицы, строки которой соответствуют состояниям, а столбцы – символам алфавита. В каждой ячейке таблицы задан символ, который машина записывает на ленту, состояние, в которое она переходит, и направление движения головки. Такую таблицу будем называть *таблицей переходов*.

В качестве примера построим машину Тьюринга, прибавляющую единицу к данному двоичному числу. Для того чтобы это сделать, надо сначала подвести головку машины к крайнему правому символу (это будет производиться в состоянии q_0), затем прибавить единицу (это будет происходить в состоянии q_1) и далее перейти в допускающее состояние q_f . Таблица переходов этой машины Тьюринга приведена ниже (табл. 1).

Таблица 1

Таблица переходов машины Тьюринга, прибавляющей единицу к заданному числу

Состояние	Символ		
	0	1	—
q_0	0, q_0 , R	1, q_0 , R	_, q_1 , L
q_1	1, q_f , S	0, q_1 , L	1, q_f , S
$q_f \in F$			

Теперь построим машину Тьюринга, допускающую язык $\{0^n 1^n \mid n \geq 1\}$. На ее вход подается слово над алфавитом $\{0; 1\}$. Начиная с левого конца входного слова, машина Тьюринга заменяет символ «0» символом «A», затем передвигается вправо, пока не достигнет символа «1». Его она заменяет символом «B» и движется

ОГЛАВЛЕНИЕ

Введение	3
1. Основные понятия	4
2. Алгоритмически неразрешимые задачи	10
3. Классы сложности	15
4. NP -полнота	16
5. Схемы из функциональных элементов	23
6. Вероятностные алгоритмы	32
7. Защита информации и теория алгоритмов	34
Задачи для самостоятельного решения.....	35
Литература.....	37