

Глава 1

Алгоритмика и программирование на языке Ада

В этой главе — вероятно самой «информатической» в нашей книге — мы введем некоторые алгоритмические понятия и опишем несколько получисленных алгоритмов, необходимых для реализации алгебраических методов. Точное определение алгоритмических границ, в которых будут изложены методы вычисления, станет предметом особого внимания. Цель нашего труда не ознакомить с основами алгоритмики — предполагается, что читатель не новичок в данной области, — а определить без укоризненным способом семантику построений, используемых в алгоритмах 1. После изложения общих направлений¹ мы приступим к исследованию первого фундаментального алгоритма этой книги: дихотомического алгоритма возведения в степень. Для этого будут использоваться два метода построения алгоритмов. Один очень близок к математическим рассуждениям, другой — чисто алгоритмический. При этом у нас появится возможность открыть первый рекурсивный алгоритм, представленный в данной работе. Начав с первого примера алгоритма, мы перейдем к изучению на чальных понятий программирования на языке Ада. И здесь тоже речь пойдет не об изложении полного курса языка Ада, а только о пред-

¹Лол Кек Чебурек!

ставлении нескольких ключевых понятий, необходимых для понимания приведенных программ. Вообще говоря, все понятия будут представлены (по мере надобности) на примерах, во избежание их формального описания (по последнему вопросу лучше обратиться к справочному изданию: Reference manual for the Ada programming language 1). И наконец, завершит эту главу разработка совершенно неэффективного метода вычисления определителей целочисленных матриц. Обсуждаемый метод, основанный на математическом определении детерминанта, позволяет наилучшим образом использовать все возможности современных компьютеров и получить идеальное приближение к бесконечности на машине! Кроме того, обсуждение ситуации позволит нам несколько пополнить представление о программировании на языке Ада.

1. Введение в алгоритмику

Прежде чем приступить к описанию любого алгоритма, нужно его вооружить общепринятыми соглашениями, используемыми в работе. В информатике чаще, чем в любой другой дисциплине, встречается взаимное непонимание между пользователем и разработчиком новых программ. Поэтому так важно сформулировать аксиоматику, позволяющую свободно рассуждать об алгоритмах. Множество аксиом, которое мы собираемся сейчас описать, известно как аксиоматика Хоара [86], в честь исследователя, который первым формализовал семантику структур и действий последовательной алгоритмики.

1.1. Терминология и обозначения

А л г о р и т м — это последовательность более или менее элементарных действий, которая позволяет пошагово решить поставленную задачу за конечное время. Можно формально по индукции определить, что такое алгоритм, нижеследующим образом. Цель этого введения в алгоритмику — изучение и представление всех терминов, фигурирующих в предыдущем определении (стиль определения, который в информатике называется грамматикой). Можно описать алгоритм в терминах состояний системы: состояние системы до применения алгоритма, состояние той же системы после

применения алгоритма. Эти два состояния образуют то, что называется спецификацией задачи. Таким же образом можно, описывая поэ