Вместо предисловия

> Чиновники от образования все менее и менее желают, чтобы молодежь становилась образованной. Их задача — не развивая умственных способностей, просто сообщать информацию. (Вероятно, именно с этим связано и увлечение тестированием, так как именно усвоение информации легко и просто проверяется с помощью тестов.) В результате мы сталкиваемся с тем парадоксальным фактом, что образование становится одним из основных препятствий к развитию интеллекта и свободе мысли.
>
> *Б. Рассел*

Мы в своих книгах постоянно говорим об интеллекте, о развитии интеллекта. Не слишком ли это претенциозно?

Возможно, мы вкладываем в данное понятие нечто другое, не то, о чем говорят философы, психологи, и на основании нашего определения рассуждаем об интеллекте. Действительно, например, в психологии достигнуты впечатляющие результаты. Помимо того что научились оценивать уровень развития интеллекта, последний еще и классифицирован. «В науке известно 120 видов интеллекта»[1]. Из наиболее часто встречающихся называют интеллект: физический, возрастной, чувственный, сексуальный, творческий, социальный, личностный, духовный[2]. 120 видов интеллекта происходят из тестологической теории интеллекта. Но она не единственная. Перечислим некоторые из теорий: теория интеллекта Ж. Пиаже; культурно-историческая теория интеллекта Л. С. Выготского; процессуально-деятельностные теории интеллекта (С. Л. Рубинштейн, П. Я. Гальперин, А. В. Брушлинский, Л. А. Венгер, О. К. Тихомирова); функционально-уровневые теории интеллекта (Б. Г. Ананьев, Б. М. Величковский); регуляционные теории интеллекта (Л. Л. Терстоуном, Р. Стернберг) и т. д. Общим знаменателем перечисленных теорий является то, что они построены на *индуктивных началах*. Другими

---

1 *Кинякина О. Н.* и др. Мозг на 100%. Интеллект. Память. Креатив. Интуиция. Интенсив-тренинг по развитию суперспособностей. М.: Эксмо, 2009, С. 75.

2 Там же. С. 75.

словами, теоретические построения основаны на обобщении экспериментальных, практических результатов[1].

Есть и другой путь — *дедуктивный*. Л. М. Веккер (онтологическая теория интеллекта) писал «о неправомерности описания психической реальности через совокупность её свойств»[2]. М. А. Холодная говорит о структурно-интегративной методологии, которая «означает принципиальную смену исследовательской парадигмы, а именно: переход от описательного уровня анализа свойств интеллекта, с высокой степенью вариативности и разнообразия обнаруживающих себя в условиях тех или иных «задачных» ситуаций, к объяснительному уровню анализа этих свойств за счет выявления структурной организации интеллекта, по отношению к которым эти интеллектуальные свойства выступают в качестве производных»[3].

Итак, пусть мы придерживаемся дедуктивной схемы рассуждений. Но мало придерживаться, т. е. определить нечто как интеллект, постулировать некие аксиомы и вывести логически свойства, особенности обозначенных явлений. Да, мы можем определить интеллект, в первую очередь, как развитые аналитические свойства ума. Но он состоит не только в этом, и не будем торопиться. Например, после творческой проработки материала данной книги у вас возникнет нечто, называемое интуицией. При «встрече» с новой проблемой (задачей) вы будете уже интуитивно чувствовать, знать, решается она или нет методом динамического программирования.

В соответствии с синергетической парадигмой мы определяем интеллект как сложную, открытую, самоорганизующуюся систему, развивающуюся по нелинейным законам (*A*). Не будем пока обосновывать и раскрывать весь тот позитив, который уже дает такое понимание, а выстроим канву обоснования исходного утверждения. Необходимым условием развития интеллекта является помещение интеллекта в среду, опять же сложную, открытую, функционирующую по нелинейной динамике (*B*), но и это еще не всё. Деятельность интеллекта по

---

[1] Индуктивные построения в любой науке должны быть, они содержат как объяснительный, так и прогнозирующий потенциал.

[2] *Веккер Л. М.* Психические процессы. Мышление и интеллект. Т. 2. Л.: Изд-во Ленингр. ун-та, 1976.

[3] *Холодная М. А.* Психология интеллекта. Парадоксы исследования. СПб.: Питер, 2002. С. 81.

«выживанию» в данной среде, в свою очередь, подчиняется тем же законам — она сложна, открыта и нелинейна (*C*). Что мы имеем в результате? В математике в этом случае говорят как минимум о том, что есть взаимно однозначное соответствие между *A*, *B* и *C*. Мы говорим о синергетической среде обучения или среде по «произрастанию» интеллекта.

Обоснование и раскрытие положения *A* очень не просто. Появление работ по философии, психологии, в полной мере доказывающих эту установку и показывающих, к чему она приводит, следует только приветствовать. Мы сошлемся на мысль К. Майнцера[1], который пишет: «Наш подход предполагает, что физическая, социальная и *ментальная реальность* является нелинейной и сложной... Этот существенный результат синергетической эпистемологии влечет за собой серьезные следствия для нашего поведения. Стоит еще раз подчеркнуть, что линейное мышление может быть опасным в нелинейной сложной реальности... Наши врачи и психологи[2] должны научиться рассматривать людей как сложных нелинейных существ, обладающих умом и телом. Линейное мышление может терпеть неудачу в установлении правильных диагнозов... Мы должны помнить, что в политике и истории монокаузальность может вести к догматизму, отсутствию толерантности и фанатизму... Подход к изучению сложных систем порождает новые следствия в эпистемологии и этике. Он дает шанс предотвратить хаос в сложном нелинейном мире и использовать креативные возможности синергетических эффектов»[3].

Положение *C* раскрыто в работе одного из авторов[4]. В ней показано, что деятельность по разработке даже простой миниатюрной программы носит нелинейный характер. Таким образом, речь идет не просто об изучении информатики по учебникам, а о работе с предметом «Информатика» путем (через, посредством) создания и анализа программных решений, и этот вид деятельности является основным.

---

[1] Президент Немецкого общества по изучению сложных систем и нелинейной динамики.

[2] Мы бы добавили: и педагоги.

[3] *Майнцер К.* Сложносистемное мышление. Материя, разум, человечество. Новый синтез. М.: Книжный дом «Либроком»/URSS, 2009.

[4] *Окулов С. М.* Информатика: развитие интеллекта школьников. 2-е изд. М.: БИНОМ. Лаборатория знаний, 2008.

Рассмотрим положение *В*. Компонентом среды является содержание предмета, и именно *часть этого содержания представлена в данной книге*. Диапазон представления достаточно обширен — от простых проблем до достаточно сложных, но главное в том, что, несмотря на кажущуюся сложность проблем, авторы пытались показать те простые идеи, положения, которые лежат в основе их решения. В указанной работе одного из авторов раскрывается суть сложности и открытости содержания, а это необходимое условие, если так можно выразиться, его синергетичности. Однако при всей открытости содержания принцип его фундаментальности является основным, ибо образование, особенно школьное, не может быть флюгером, отрабатывающим последние и быстропроходящие новомодные веяния (особенно в информатике).

Мы не будем раскрывать основные моменты по созданию синергетической среды обучения, они известны[1]. Напомним только, что речь не идет о простом школьном курсе по информатике (он, если так можно выразиться, лишь один из элементов среды). Среда — это большее, тем более что она находится в определенном противоречии с традиционной классно-урочной системой изучения предмета. Эта система насквозь дуальна. Ученик, по образному выражению К. Поппера, рассматривается «как сосуд по вливанию жидкости (знаний)». Уход от дуализма в образовании кажется утопией, но известные «площадки» (г. Санкт-Петербург, г. Саратов, г. Новосибирск, г. Казань, г. Мытищи Московской области, СУНЦ МГУ г. Москва), в которых созданы определенные среды (каждая имеет свое, специфическое «лицо», но построены они на единых принципах) по развитию интеллекта школьника через изучение информатики, вселяют надежду.

[1] Окулов С. М. Информатика: развитие интеллекта школьников. 2-е изд. М.: БИНОМ. Лаборатория знаний, 2008.

Введение

> Новое по самому своему определению — это преходящая сторона вещей... Самое лучшее в новом то, что отвечает старому устремлению.
>
> *П. Валери*

> Природа нового парадоксальна: ничто не ново в этом открытом креативном, т. е. постоянно творящем новое, мире.
>
> *Е. Н. Князева, С. П. Курдюмов*

В историческом плане понятие «динамическое программирование» было введено Ричардом Беллманом (Richard Bellman) в 1950-х годах и определяло раздел прикладной математики под названием «исследование операций». Круг исследуемых задач был достаточно четко обозначен. Это методы поиска оптимальных (наилучших) решений в задачах управления системами, но с одной, если так можно выразиться, особенностью. Как изменение самой системы во времени, так и процесс управления системой допускал разбивку по времени на этапы (шаги). Отсюда и термин «динамическое» — изменяющееся во времени. Но так можно представить (описать) практически любую систему управления. Особенность динамического программирования заключалась в том, что допускалась разбивка процесса на фиксированные промежутки времени и в целом оптимальное решение задачи как бы складывалось из оптимальных решений на каждом из промежутков (этапе, шаге). Термин «динамическое программирование» (dynamic programming) в данном случае никоим образом не связан с разработкой программ для компьютера, так же как, например, и «линейное программирование» (linear programming). Он означает нечто другое, а именно строго заданную последовательность операций (арифметических, логических) по нахождению оптимального решения. Фактически это алгоритм решения задачи.

В ходе дальнейшего развития, но уже не раздела прикладной математики, а информатики в целом с понятием «динамическое программирование» произошла некая трансформация. Оно очерчивает вполне определенный метод проектирования алгоритмов, который не ограничивается только задачами оптимизации. Естественно, этот метод не универсален, он работоспособен для вполне определенного класса задач. В идее разбивки задачи на подзадачи и компоновки решения задачи из решений подзадач нет ничего нового — это универсальный

метод. Смысл (суть) заложен в интерпретации терминов «разбивка» и «компоновка». Задача должна допускать разбивку на подзадачи того же вида (типа) так, чтобы её решение как бы складывалось, компоновалось из решений подзадач. Другими словами, должны быть выведены (определены) функциональные зависимости (рекуррентные соотношения) между решениями подзадач. Тогда, начиная, например, с небольших задач, мы постепенно получаем решения все больших задач и наконец доходим до решения исходной задачи. Особенность динамического программирования как метода заключается и в том, что каждая подзадача решается только один раз. Результат её решения запоминается и затем, при решении следующих подзадач, используется как данное. Итак, два ключевых момента: «связь» и «запоминание».

Казалось бы, в чем сложность? Дана задача — применяй метод, как в случае перебора с возвратом. Но, к сожалению, а может быть к счастью, не все так просто. Универсальных рекомендаций о том, решается или нет конкретная задача с помощью динамической схемы, не разработано. Это необходимо определить, понять, что бывает сделать в определенных случаях достаточно сложно. И так как отсутствуют строгие правила (леммы, теоремы и так далее), это не случай теоремы Пифагора, то остается идти, познавая метод, только от практики решения конкретных задач.

Структура книги

В главе 1 рассмотрен ряд простых задач. О динамическом программировании не говорится. Задачи служат как бы «затравкой». Идеи метода динамического программирования используются, но детального «разговора» о них нет. С одной стороны, закладывается базис для понимания метода, а с другой, появляется материал, на который мы имеем право ссылаться в последующем изложении.

В главе 2 излагаются общие положения метода и показываются способы его реализации.

В главе 3 задачи, решаемые методом динамического программирования, классифицируются. Эта классификация идет «от практики» работы со школьниками, она не встречается в учебной литературе. Вероятно, что сделана первая попытка такой работы, поэтому авторы сознательно подставляют себя под критику (но «не кусайте» слишком сильно).

В приложении I дается материал по классическому способу изложения метода динамического программирования для студентов в рамках соответствующих курсов.

Приложение II посвящено обзору задач по данной проблематике, которые встречаются в учебной литературе. Степень разбора задач различна. В некоторых случаях она вполне достаточна не только для полного понимания (это предполагается), но и для проведения «полнокровного» изложения материала.

Необходимые условия для работы с книгой

Основного курса информатики, проводимого через программирование[1] как вида основной деятельности на занятиях, достаточно для понимания и свободного освоения материала данной книги.

В заключение отметим, что задачи, связанные с данной проблематикой, встречаются практически на каждой олимпиаде по информатике (и являются, соответственно, одним из разделов подготовки школьника к этим мероприятиям), начиная, вероятно, с 1993 года (Международная олимпиада школьников по информатике в Аргентине), но изложение соответствующего материала в учебной литературе носит фрагментарный характер. Алгоритмы (основные) входят в примерную программу по олимпиадной информатике под названием «динамическое программирование»[2].

Алгоритмы динамического программирования относятся к дидактическим единицам, «изучение которых формирует у школьников ключевые умения в области олимпиадной подготовки, открывает перед участником олимпиадного состязания возможность проявить свой творческий потенциал на достойном уровне... — дипломов победителей и призеров заключительных этапов Всероссийской олимпиады школьников»[3].

---

[1] Например, на основе первых частей книги: *Окулов С. М.* Основы программирования. 5-е изд. М.: БИНОМ. Лаборатория знаний, 2010.

[2] *Кирюхин В. М.* Информатика: всероссийские олимпиады. М.: Просвещение, 2008. С. 72.

[3] Там же. С. 67.

Глава 1

Простые задачи

Дума за думой, волна за волной —
Два проявления стихии одной:
В сердце ли тесном, в безбрежном ли море,
Здесь — в заключении, там — на просторе —
Тот же всё вечный прибой и отбой,
Тот же всё призрак — тревожно-пустой.

Ф. И. Тютчев

В главе рассматриваются простые задачи — это первая «волна» содержания. Затем пойдет вторая волна, и т. д. От простого — к сложному, от сложного — вновь к простому. Однако к окончанию работы с книгой (естественно, с разработкой программ и их тестированием) любой «шторм» из задач по динамическому программированию вам будет не страшен.

1.1. Числа Фибоначчи[1]

Человек весь состоит из вопросов, а жизнь и окружающий мир — из ответов на эти вопросы. Определи последовательность занимающих тебя вопросов, начиная с самых важных. Потом настройся на то, чтобы воспринять ответы. Они повсюду — во всяком событии, во всякой вещи.

Борис Акунин

Знаменитая последовательность чисел 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89,... отличается тем свойством, что каждое очередное число является суммой двух предыдущих. Обозначим n-е ($n \geq 0$) число в последовательности как F_n (в приведенном ряде чисел $F_{11} = 89$). Тогда последовательность определяется следующим рекуррентным соотношением: $F_n = F_{n-1} + F_{n-2}$ ($n > 1$) и начальными значениями $F_0 = 0$, $F_1 = 1$.

---

[1] Считается, что эти числа ввел в 1202 г. Леонардо Фибоначчи (Leonardo Fibonacci), но только после работ математика Э. Люка, жившего в XIX веке, это название – «числа Фибоначчи» стало общепринятым. Упоминание об этих числах есть в работах индийских математиков: Гопала (Gopala) — до 1135 г. и Хемачандра (Hemachandra) – в 1150 г.

Сформулируем задачу: по заданному числу n вычислить F_n.

Примечание. Использование известной формулы Ж. Бине:

$$F_n = \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^n + \left(\frac{1-\sqrt{5}}{2}\right)^n\right)$$

и её сокращенного варианта (ввиду того что второе слагаемое экспоненциально убывает):

$$F_n\left\lfloor \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n \right\rfloor$$

требует использования вещественной арифметики повышенной точности. Этот вариант решения не относится к рассматриваемой проблематике[1].

Рекурсивный вариант решения задачи очевиден для чисел, «укладывающихся» в диапазон LongInt. Ограничимся этим диапазоном, для больших значений требуется использовать специальные приемы выполнения арифметических операций[2].

```
Function Fib(n: LongInt): LongInt;
  Begin
    If n<2 Then Fib:=n
      Else Fib:=Fib(n-1)+Fib(n-2);
  End;
```

Логика решения проиллюстрирована на рис. 1.1, она имеет экспоненциальную временну́ю сложность — $O(2^n)$.

Рассмотрим другой вариант решения задачи. Определим понятие подзадачи. В данном случае это вычисление очередного числа Фибоначчи. Для каждого i, зная результаты решения подзадач для $i-1$ и $i-2$, т. е. числа F_{i-1} и F_{i-2}, мы решаем очередную подзадачу. И так до значения n. При этом каждая подзадача решается только один раз. Другими словами, мы

---

[1] Если число Фибоначчи входит в стандартный диапазон LongInt (Int64), то никаких проблем с вещественными числами нет. Мы можем использовать стандартную функцию любого языка программирования — возведение в степень и затем округление.

[2] *Окулов С. М.* Программирование в алгоритмах. — 3-е изд. — М.: БИНОМ. Лаборатория знаний, 2007. С. 9–24.

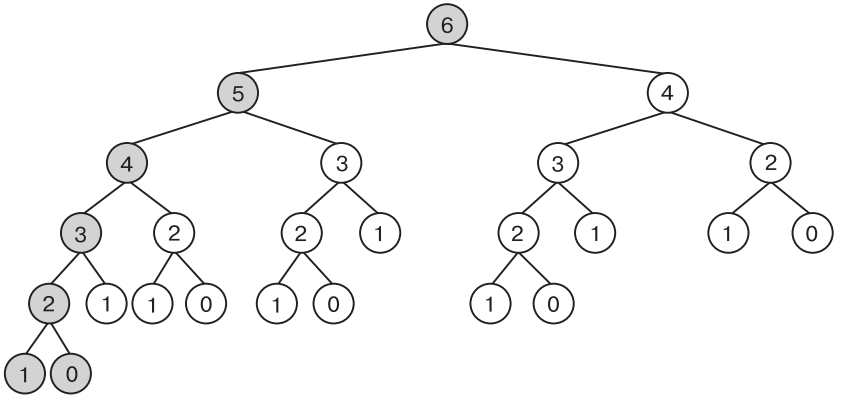

Рис. 1.1. Последовательность вызовов функции Fib в процессе вычисления F_6

«проходим» только по «кружкам», выделенным темным цветом на рис. 1.1. Предположим, что результаты решения подзадач хранятся в массиве *F* (табл. 1.1).

Таблица 1.1

i	0	1	2	3	4	5	6	7	8	9	10	11	...
F[*i*]	0	1	1	2	3	5	8	13	21	34	55	89	...

Запись логики решения имеет следующий вид.

```
Procedure FibD(n: LongInt);
  Var i: LongInt;
  Begin
    F[0]:=0;
    F[1]:=1;
    For i:=2 To n Do F[i]:=F[i-1]+F[i-2];
    WriteLn(F[n]);
  End;
```

Так как для решения очередной подзадачи требуются результаты решения только двух предыдущих подзадач, то массив *F* излишен, достаточно двух переменных.

```
Procedure FibD(n: LongInt);
  Var i,a,b,c: LongInt;
  Begin
    a:=0;
    b:=1;
    For i:=2 To n Do Begin
      c:=a+b;
      a:=b;
      b:=c;
    End;
    WriteLn(c);
End;
```

Временна́я сложность вычислений — $O(n)$.

Примечание

Существует метод вычисления чисел Фибоначчи с временно́й сложностью $O(\log_2 n)$. Запишем соотношение: $F_n = F_{n-1} + F_{n-2}$ в матричной форме:

$$\begin{pmatrix} F_{n-1} \\ F_n \end{pmatrix} = A \cdot \begin{pmatrix} F_{n-2} \\ F_{n-1} \end{pmatrix}, \text{ где } A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}.$$

Тогда

$$\begin{pmatrix} F_n \\ F_{n+1} \end{pmatrix} = A \cdot \begin{pmatrix} F_{n-1} \\ F_n \end{pmatrix} = A^2 \cdot \begin{pmatrix} F_{n-2} \\ F_{n-1} \end{pmatrix} = \ldots = A^n \cdot \begin{pmatrix} F_0 \\ F_1 \end{pmatrix}.$$

Для вычисления A^n можно воспользоваться алгоритмом быстрого возведения в степень с временно́й сложностью $O(\log_2 n)$[1]. Считается, что пара матриц размера 2×2 перемножается за постоянное время.

Итак, причем здесь динамическое программирование? Мы определили понятие «подзадача», мы показали, как из решения подзадач «складывается» решение исходной задачи. Каждую подзадачу мы решали только один раз, запоминая результаты в массиве. В результате этих действий получен выигрыш по времени и первоначальное решение с экспоненциальной временно́й сложностью трансформировалось в решение с линейной сложностью и даже более эффективное, чем с линейной сложностью.

---

[1] *Шень А.* Программирование: теоремы и задачи. М.: МЦНМО, 1995. С. 8.

[. . .]