

ЗАНЯТОЙ БОБЁР

От редакции: статья посвящена обсуждению задачи «Занятой бобёр» конкурса КИО-2007.

От автора: идею о том, как устроить перебор всех программ для Бобра, мне подсказал друг Дмитрий Михайлович Ицыксон, аспирант в ПОМИ, он как раз занимается теорией сложности. За что я ему очень благодарен.

В задаче требуется написать программу, которая управляет поведением Бобра. Он передвигается вдоль «бесконечной в обе стороны» реки и совершает простые действия: кладет бревно в реку (если его там нет), вынимает бревно из реки (если оно там есть) или не делает ни того, ни другого.

При совершении действий Бобёр изменяет свое состояние. Всего состояний пять. Перед началом работы Бобёр «воодушевлен» – это его начальное состояние. Если Бобёр становится «усталым», он заканчивает работу – это конечное состояние. Также Бобёр может быть «бодр», «взъерошен» и «доволен».

Программа управления Бобром состоит из набора инструкций, пример возможной инструкции выглядит так: «Если Бобёр воодушевлен и перед ним нет бревна, то он кладет бревно, становится бодр и идет вправо». Из примера видно, что Бобёр выбирает свои действия в зависимости от своего текущего состояния, а также от того, лежит перед ним бревно или нет. Само действие состоит из трех – сначала Бобёр «исправляет» содержимое реки перед собой, возможные варианты – положить бревно, убрать бревно, оставить все как есть. Потом Бобёр изменяет свое состояние на одно из пяти возможных, и, наконец, перемещается: он может сделать шаг направо, налево,

а также остаться на месте. Можно привести еще один пример инструкции: «Если Бобёр бодр и перед ним лежит бревно, то он оставляет как есть, становится усталым и идет влево».

Перед исполнением программы вся река полностью пуста и не содержит бревен, Бобёр стоит на берегу в состоянии «воодушевлен». На каждом шаге исполнения выбирается инструкция в зависимости от текущего

состояния Бобра и того, лежит ли перед ним бревно. Если соответствующей инструкции не нашлось, программа останавливается и Бобёр завершает свою работу. Аналогично, программа останавливается, если Бобёр на очередном шаге оказался в состоянии «усталый». Программа может

быть написана неудачно, и окажется, что Бобёр будет работать бесконечно. Программа для Бобра отличается от программ на языках программирования типа Pascal и Basic, тем, что в ней не важен порядок инструкций.

На конкурсе предлагалось два варианта условия задачи. В первом варианте требовалось написать программу, при которой Бобёр сделает *как можно больше шагов* (считаются шаги влево и вправо) и остановится. Программы, при которых Бобёр не останавливается, не являются решениями. Во втором варианте необходимо было на-

На конкурсе предлагалось два варианта условия задачи. В первом варианте требовалось написать программу, при которой Бобёр сделает как можно больше шагов (считаются шаги влево и вправо) и остановится. Во втором варианте необходимо было написать программу, при которой Бобёр положит в реку как можно больше бревен.

писать программу, при которой Бобёр *положит в реку как можно больше бревен*. Аналогично, учитываются только программы в которых Бобёр в конце концов останавливается.

Далее мы рассмотрим результаты, которые получали участники, а также выясним максимальный результат, который теоретически можно было получить. Но сначала мы скажем несколько слов о содержании задачи.

МАШИНА ТЬЮРИНГА

Задача «Занятой Бобёр» связана с одной из самых распространенных формализаций понятия алгоритма – Машиной Тьюринга. Бобёр в данной задаче является частным случаем Машины Тьюринга. В общем случае у нее может быть не только 5, но любое конечное число состояний. К тому же, на ленте (реке) могут находиться символы из некоторого множества, не обязательно множества «есть бревно» / «нет бревна». Машина Тьюринга была результатом решения Аланом Тьюрингом проблемы поиска точного математического эквивалента интуитивного понятия «алгоритм». Действительно, сейчас считается что все, что может быть вычислено с помощью «алгоритмов» в широком интуитивном смысле (то есть некоторой четко заданной последовательности действий), может быть также вычислено и на Машине Тьюринга. Писать программу для Машины Тьюринга значительно сложнее, чем, например, писать программу на Pascal, но Машины Тьюринга устроены настолько просто, что именно их используют для изу-



...требуется положить в реку как можно больше бревен.

чения алгоритмов. В частности для изучения скорости работы алгоритмов и принципиальной возможности создать какой-либо алгоритм. Так, например, можно доказать, что не существует алгоритма (читай, Машины Тьюринга), который мог бы по любой Машине Тьюринга сказать, остановится ли она, если начнет работу при пустой ленте (ситуация как в задаче – Бобёр начинает работу с пустой рекой без бревен).

Получился интересный результат, если мы написали программу для Бобра, мы не можем так просто определить, остановится он или нет. Если запустить Бобра работать и увидеть, что он уже сделал 100 шагов, то непонятно, он сейчас сделает еще несколько шагов и остановится, или, что более вероятно, будет продолжать работать бесконечно. Мы можем дождаться, когда Бобёр сделает 1000 шагов и надеяться, что 1001 шаг будет последним, но вероятнее Бобёр так никогда и не остановится. В некоторых случаях можно определить, что Бобёр зациклился, пример этой ситуации – Бобёр все время шагает направо, не изменяя состояние.

РЕШЕНИЯ УЧАСТНИКОВ

ВАРИАНТ С НАИБОЛЬШИМ КОЛИЧЕСТВОМ БРЕВЕН

Рассмотрим для начала решение задачи, в которой требуется положить в реку как можно больше бревен. В таблице приведены результаты, сколько участников и какое количество бревен смогли положить (результаты менее 4 бревен не указаны):

Бревен	Решивших
4	36
5	22
6	6
7	17
8	2

Практически все присланные программы для Бобра различаются, и трудно найти в них что-либо общее. Но все же, давайте посмотрим, как можно целенаправленно получить на реке 4 и более бревен. Приво-

димые ниже программы в том или ином виде встречаются и у участников.

Дальнейшие рассуждения для увеличения их наглядности можно проверять с помощью программы «Занятой Бобёр». В начальный момент Бобёр стоит перед пустой рекой в состоянии «воодушевлен». Зададим ему инструкцию: (1) **Если Бобёр воодушевлен и перед ним нет бревна, то он кладет бревно, становится бодр и идет вправо.** После исполнения этой инструкции Бобёр будет стоять перед пустой рекой, но уже в состоянии «бодр». Напишем аналогичную инструкцию и переведем Бобра в следующее состояние: (2) **Если Бобёр бодр и перед ним нет бревна, то он кладет бревно, становится взъерошен и идет вправо.** Аналогично положим еще одно бревно и переведем Бобра в следующее состояние: (3) **Если Бобёр взъерошен и перед ним нет бревна, то он кладет бревно, становится доволен и идет вправо.** Наконец, положим последнее, четвертое бревно. Так как состояния закончились, программе, видимо, придется завершиться: (4) **Если Бобёр доволен и перед ним нет бревна, то он кладет бревно, становится усталым и идет вправо.**

Бобёр, руководствуясь приведенной программой, положит 4 бревна и остановится. Как улучшить программу и положить 5 бревен? Изменим инструкцию (4), Скажем Бобру, чтобы он, во-первых, не уставал после 4-ого положенного бревна, а во-вторых, скажем ему остаться на месте. Итак, исправленная (4) инструкция: (4*) **Если Бобёр доволен и перед ним нет бревна, то он кладет бревно, становится воодушевлен и остается на месте.** (Вместо «воодушевлен» можно было выбрать и другое какое-нибудь состояние). После исполнения инструкции (4*) перед воодушевленным Бобром будет

лежать бревно, самое правое из четырех уже положенных (рис. 1).

Научим Бобра идти влево, пока перед ним лежат бревна: (5) **Если Бобёр воодушевлен и перед ним есть бревно, то он оставляет как есть, становится воодушевлен и идет влево.** Так Бобёр пройдет все 4 бревна, пока не встанет слева от них. Далее, в состоянии «воодушевлен» он будет стоять перед пустой рекой. Что делать в этой ситуации он уже знает, у него есть инструкция (1), он положит бревно, станет бодр и пойдет вправо. Для состояния «бодр», когда перед Бобром лежит бревно, программа не дает инструкций к действию, поэтому Бобёр остановится. Итак, программа из инструкций (1), (2), (3), (4*), (5) заставляет Бобра положить 5 бревен. Из результатов видно, что положить 5 бревен смогло чуть меньше участников, чем положить 4 бревна.

Чтобы положить 7 бревен, нужно дополнить нашу программу еще несколькими инструкциями, аналогичными по сути предыдущим: (6) **Если Бобёр бодр и перед ним есть бревно, то он оставляет как есть, становится бодр и идет влево.** (7) **Если Бобёр взъерошен и перед ним есть бревно, то он оставляет как есть, становится взъерошен и идет влево.** Как именно Бобёр работает в соответствии с этими инструкциями, предлагается читателю разобраться самому.

Из результатов видно, что те, кто догадался как положить 6 бревен, в большинстве своем догадались и до результата в 7 бревен, действительно, результат в 6 бревен очень непопулярен, и 7 бревен получило значительно большее количество человек.

Результат в 8 бревен оказался доступен только двоим участникам. И если для того, чтобы положить 7 бревен, Бобёр делает вроде бы осмысленные действия (мы сами



Рис. 1

писали для него программу, разбираясь, как она работает), то программа для восьми бревен выглядит бессмысленной. Выполняя ее, Бобёр хаотично ходит вдоль реки, кладя и убирая (!) бревна. Заметим, что в программе для 7 бревен не было ни одной инструкции, предполагающей убирание бревна. Возможно, только сами авторы решений с восьмью бревнами понимают, как работает их программа.

Приведем решения участников, заставляющие Бобра положить 8 бревен. Как видно, оба решения абсолютно разные. Для экономии места приведем решения в таблицах. «ПБ» – положить бревно, «УБ» – убрать бревно, «Ост» – Оставить как есть. Стрелка означает направление движения, отсутствие стрелки – остаться на месте.

	Есть бревно	Нет бревна
Воод.		ПБ, бодр, →
Бодр	Ост, довол., ←	Ост, взърш., →
Взърш.	УБ, воод., ←	ПБ, довол., →
Довол.		ПБ, бодр, ←

Второе решение:

	Есть бревно	Нет бревна
Воод.		ПБ, бодр, →
Бодр	Ост, бодр, ←	ПБ, взърш., →
Взърш.	УБ, бодр, →	ПБ, довол.
Довол.	Ост, довол., ←	ПБ, воод., ←

**ВАРИАНТ
С НАИБОЛЬШИМ КОЛИЧЕСТВОМ ШАГОВ**

Результаты участников сведены в таблицу. Количество шагов от 5 до 12 получи-



Бобёр хаотично ходит вдоль реки, кладя и убирая (!) бревна...

ло примерно одинаковое число человек, они объединены в одной строке:

Шагов	Решивших
18	6
15	2
13	2
5 – 12	от 3 до 7
4	15

Чтобы заставить Бобра сделать 4 шага, достаточно написать программу из инструкций (1), (2), (3), (4), которую мы раньше использовали, чтобы положить 4 бревна. Можно заставить Бобра сделать 8 шагов, для этого необходимо воспользоваться программой из инструкций (1), (2), (3), (4*), (5), которую мы использовали ранее, чтобы положить 5 бревен. Правда, результат можно улучшить до 9 шагов, необходимо добавить инструкцию (8) **Если Бобёр бодр и перед ним есть бревно, то он оставляет как есть, становится усталым и идет влево.** Теперь Бобёр сделает еще один лишний шаг перед тем как закончить работу.

Программа из предыдущего раздела для 7 бревен заставит Бобра сделать 14 шагов. Если улучшить ее также, как программу для 5 бревен, заставив Бобра сделать дополнительный лишний шаг в самом конце, то получится результат в 15 шагов. Это решение было прислано одним из участников.

Решения для 18 шагов заставляют Бобра действовать хаотически и без видимой цели, он ходит влево и вправо, кладя и убирая бревна. Впрочем, практически все решения этой задачи, заставляющие Бобра сделать как можно больше шагов, имеют такой вид. Если кто-то и понимает, как устроено конкретное решение, то только сам автор.

Итак, одно из решений для 18 шагов:

	Есть бревно	Нет бревна
Воод.	Ост., довол., ←	ПБ, бодр, →
Бодр	УБ, устал., ←	ПБ, взърш., →
Взърш.	УБ, бодр, ←	ПБ, довол., →
Довол.	УБ, взърш., ←	ПБ, воод., ←

Решения из предыдущего раздела, кладущие по 8 бревен, заставляют Бобра сделать соответственно 30 и 28 шагов (реше-

ния необходимо подправить, заставив Бобра сделать лишний шаг перед окончанием работы).

КАКИЕ МАКСИМАЛЬНЫЕ ОТВЕТЫ МОЖНО БЫЛО ПОЛУЧИТЬ

Задача устроена таким образом, что в ней трудно проводить какие-либо математические рассуждения и доказывать про нее утверждения. Например, непонятно, как оценить наибольшее возможное количество шагов Бобра или наибольшее возможное количество бревен. Вопрос состоит даже не в том, чтобы найти это наибольшее количество шагов или бревен, а хотя бы просто показать, что количество шагов не может быть больше, чем, например, 100. Вроде бы правдоподобное утверждение, особенно если судить по результатам участников: максимальное полученное число шагов – 30.

Если не получается найти математических рассуждений, можно попробовать перебрать *все* возможные программы для Бобра и найти из них оптимальную. Задача не кажется такой уж невыполнимой, если оценить количество программ, которые можно дать Бобру, то их окажется около $31^8 \approx 850$ млрд. (31 – количество вариантов того, что можно написать в клетке таблицы с решением, 8 – общее количество клеток). Если один вариант программы перебирается достаточно быстро, то 850 млрд. вариантов – вполне приемлемый размер перебора для современной вычислительной техники.

При переборе вариантов встает уже обсуждавшаяся проблема, как определить, остановится Бобёр, работающий в соответствии с некоторым очередным алгоритмом, или нет. Здесь можно ограничиться правдоподобным предположением: если Бобёр выполнил уже 1000 итераций, то, он, вероятнее всего, не остановится, и программу

можно отбросить, считая ее не подходящей под условие.

Следующая идея помогает сократить перебор и сделать его проведение вопросом не нескольких дней, а нескольких минут (возможно, часа). При переборе не следует перебирать последовательно *все* программы, достаточно строить перебираемые программы постепенно. Например, в начальный момент Бобёр в состоянии «воодушевлен» стоит перед пустой рекой. Пока что нет инструкции, которая говорит ему, что делать в этой ситуации, и необходимо *расщепить* вычисления, перебрав все возможные варианты инструкций для этой ситуации. В каждом из подвычислений Бобёр уже знает, как себя вести в ситуации «воодушевлен» и стою перед пустой рекой, расщеплять вычисления придется в других, еще не определенных ситуациях.

Реализованный таким образом перебор дает следующий оптимальный для Бобра алгоритм:

	Есть бревно	Нет бревна
Воод.	Ост, бодр, →	ПБ, бодр, ←
Бодр	УБ, взърш., →	ПБ, воод., →
Взърш.	Ост, довол., →	ПБ, устал., →
Довол.	УБ, воод., ←	ПБ, довол., ←

При этом алгоритме кладется и наибольшее количество бревен (**13**), и совершается наибольшее количество шагов (**107**). Как видно, правдоподобное предположение, что Бобёр вряд ли сможет сделать более 100 шагов, оказалось неверным. К сожалению, так и осталось предположением, что Бобёр не может совершить более 1000 итераций алгоритма, не заиклившись. Это использовалось при переборе, поэтому, возможно, не эта программа является оптимальной. Правда, в этом случае при работе по оптимальному алгоритму перед тем как остановиться Бобёр совершит более 1000 итераций.

*Посов Илья Александрович,
аспирант математико-
механического факультета СПбГУ.*



Наши авторы, 2007
Our authors, 2007