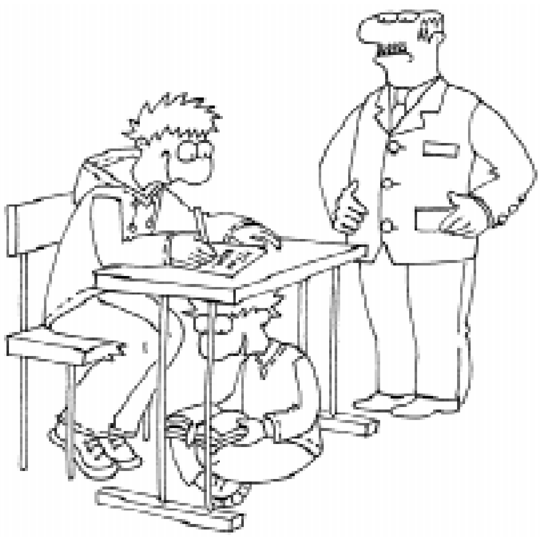
Посов Илья Александрович

АВТОМАТИЧЕСКАЯ ГЕНЕРАЦИЯ ЗАДАЧ

Преподаватель в школе или университете должен уметь проверять, какие знания присутствуют у его учеников. Для этого хорошо подходят контрольные или самостоятельные работы. Ученикам выдается некоторое количество задач, которые они должны решить самостоятельно, но обеспечить это оказывается не так просто. Борьба преподавателей со списыванием может продолжаться бесконечно, при этом совершенствование методов борьбы ведет только к совершенствованию самих методов списывания. Здесь мы посмотрим на один конкретный метод борьбы со списыванием, который состоит в том, что каждый ученик получает свой вариант задач контрольной.

Такой метод защиты от списывания значительно уменьшает возможность списать, и он просто незаменим, если контрольная выдается на дом, то есть ученики могут свободно общаться и обсуждать задачи. Аналогично, если контрольная проводится по Интернету удаленно, то нет уверенности, что ученики по ту сторону Интернета не общаются и не обсуждают задачи.



Борьба преподавателей со списыванием...

Если в контрольной 4 задачи, а в классе или группе 30 учащихся, то требуется создать 120 вариантов задач контрольной. Это может сделать либо очень трудолюбивый преподаватель, либо же это может быть сделано автоматически. Остановимся на варианте автоматического создания вариантов контрольной.

ГЕНЕРАТОР ЗАДАЧИ

Далее будем называть *генератором* программу, которая после запуска выдает текст условия некоторой задачи и ответ к задаче. Результаты запусков генераторов должны различаться, то есть каждый раз генератор должен создавать новое условие задачи. Посмотрим на пример трех запусков генератора «Алгоритм Евклида»:

Используя алгоритм Евклида, найдите наибольший общий делитель чисел 1705 и 4862. Ответ: 11

Используя алгоритм Евклида, найдите наибольший общий делитель чисел 6244 и 3759. Ответ: 7

Используя алгоритм Евклида, найдите наибольший общий делитель чисел 6795 и 2538. Ответ: 9

Попробуем реализовать такой генератор и посмотреть, какие проблемы могут при этом возникнуть.

Во-первых, видно, что все числа из условий задач четырехзначные. Это необходимо, чтобы поставить решающих в равные условия. Во-вторых, для решения этих трех задач требуется произвести примерно одинаковое количество шагов алгоритма Евклида. В первых двух случаях – 9 шагов, в третьем случае – 7 шагов. Это также необходимо, чтобы поставить решающих в равные условия. Наконец, последняя особенность результатов генерации – получаются разнообразные ответы, которые отличаются от «неинтересного» ответа «единица».

Как реализовать генератор «Алгоритм Евклида»? Что произойдет, если для создания условия выбирать два случайных четырехзначных числа? Задачи будут получаться разной степени сложности. Кому-то повезет, и он получит задачу, где для решения надо произвести всего один шаг алгоритма. Кому-то не повезет, и ему потребуется произвести 19 шагов алгоритма. Пример таких чисел 6765 и 4181. Это, соответственно, 19-ое и 18-ое числа Фибоначчи, их НОД равен 1.

Но огромная разница в возможном количестве шагов – это не единственная неприятность, которая может появиться, если для условия задачи выбрать два случайных числа. Что в этом случае произойдет с ответами? Видимо, ответы почти всегда будут 1. Иначе говоря, два случайных числа, скорее всего, взаимно просты. Попробуем проверить это предположение. Посмотрим на пары (N, M) из двух случайных, выбранных независимо четырехзначных чисел. Что значит, что N и M *не* взаимно просты? Это значит, что у них есть либо общий делитель 2, либо общий делитель 3, либо общий делитель 4 и т. д. Можно заметить, что в 1/4 всех пар на два делятся и N, и M. Аналогично, в 1/9 всех пар и N, и M делятся на 3. В 1/16 всех пар и N, и M делятся на 4, и т. д. Таким образом, мы получили, что не более чем $x = 1/4 + 1/9 + 1/16 + ...$ часть всех пар состоит из не взаимно простых чисел N и M. В оставшихся парах N и M взаимно просты.

Вычислим x. Известно, что $1/1^2 + 1/2^2 + 1/3^2 + ... = \pi^2/6$. Следовательно, $x = \pi^2/6 - 1 \sim 0{,}64$. Итого, минимум $1 - x \sim 0{,}36$ всех пар взаимно просты. Мы оценили количество взаимно простых пар очень грубо. Оказывается, что в более чем в 36% сгенерированных задач ответом будет 1. То есть как минимум треть учеников будут иметь в своих задачах одинаковый ответ. Эксперименты на компьютере показывают, что около 60% пар четырехзначных чисел взаимно просты. Попробуйте воспользоваться предложенной только что идеей подсчета количества пар взаимно простых чисел и оценить это количество точнее, получив ответ как раз примерно около 60%.

Для написания генератора нужно бороться с обеими проблемами – разное количество шагов алгоритма Евклида при решении и неразнообразные ответы. Чтобы избавиться от проблемы разного количества шагов, напишем генератор следующим образом:

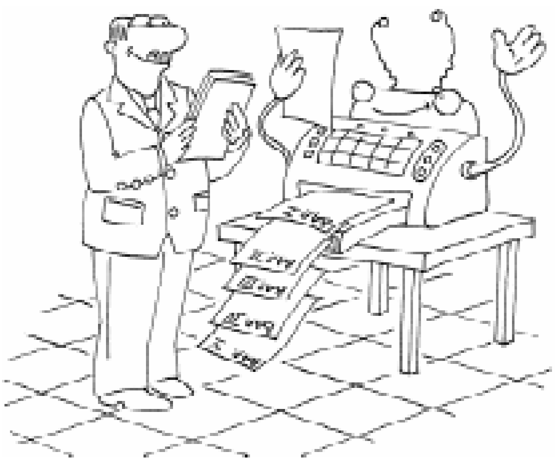
1. Выбрать два случайных четырехзначных числа M и N.

2. Найти НОД(M, N) с помощью алгоритма Евклида.

3. Если на шаге 2 в алгоритме Евклида было произведено слишком много или слишком мало шагов (например, более 10 или менее 7), вернуться к шагу 1.

4. Вывести условие и ответ.

Так как существует ненулевая вероятность, что количество шагов в алгоритме Евклида находится где-то между 7 и 10, то это событие когда-нибудь произойдет, и генератор создаст задачу. Поборемся тем



Остановимся на варианте автоматического создания вариантов контрольной.

же образом со второй проблемой, неразнообразными ответами:

1. Выбрать D – случайное число от 6 до 12. Оно будет ответом в задаче.

2. Выбрать два случайных четырехзначных числа M и N.

3. Найти НОД(M, N) с помощью алгоритма Евклида.

4. Если на шаге 3 в алгоритме Евклида было произведено слишком много или слишком мало шагов (например, более 10 или менее 7), вернуться к шагу 2.

5. Если на шаге 3 НОД(M, N) оказался не равен D, вернуться к шагу 2.

6. Вывести условие и ответ.

Если, например, выбрано $D = 6$, то в процессе работы генератора обязательно когда-нибудь найдутся два числа с НОД, равным 6. К тому же когда-нибудь для нахождения этого ответа надо будет проделать от 7 до 10 шагов.

Почему ответ в задаче D выбирается в самом начале работы генератора? Вероятность того, что два случайных четырехзначных числа имеют НОД, равный 6 отличается от вероятности, что два случайных четырехзначных числа имеют НОД, равный 7. Выбор НОД в начале работы программы гарантирует равномерность распределения ответов в условиях. Количество задач с ответом 6 будет примерно таким же, как количество задач с ответом 7.

Конечно, генератор задачи на поиск НОД может быть написан не так, как он был написан здесь. Читатель может попробовать написать генератор задачи на поиск НОД самостоятельно.

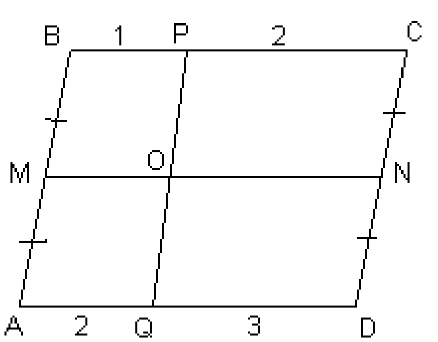


Рис. 1

ДРУГИЕ ПРИМЕРЫ ЗАДАЧ И ГЕНЕРАТОРОВ

Существует огромное количество задач, для которых теоретически может быть создан генератор. Мы посмотрим на несколько таких задач и напишем еще один генератор. Задачи возьмем из вполне определенного источника.

9 и 10 ноября 2006 года в Санкт-Петербурге проводилась городская компьютерная Интернет-олимпиада по математике для учащихся 10 и 11 классов. Участники решали задачи у себя в школах и посылали ответы по Интернету автоматической проверяющей системе. В олимпиаде каждого класса предлагалось по 10 задач, и каждый участник получал по одному из нескольких вариантов условий задания. В 10 классе участвовало примерно 700 человек, в 11 классе – примерно 900.

Решения задач проверялись автоматически, но такой вариант проведения олимпиады накладывает ограничения на типы допустимых задач. Например, задачи с требованием доказать некоторое утверждение трудно подвергнуть автоматической проверке. В задачах олимпиады, в основном, было необходимо что-либо вычислить, например, найти длину отрезка в задаче по геометрии. Автоматически проверить правильность нахождения числа не составляет проблем. В некоторых задачах ответами были списки из чисел. Список может появиться, например, в задачах на поиск корней уравнений, у которых больше одного корня. Чтобы каждый участник получил свой вариант заданий олимпиады, условия задач должны поддаваться генерации.

Какие задачи были на олимпиаде? Посмотрим на четыре задачи и потом обсудим возможность их генерации. Первая задача по геометрии:

В параллелограмме ABCD (рис. 1) *проведена средняя линия MN (M – середина AB, N – середина CD). Точка P делит отрезок BC в отношении 1:2 (считая от точки B), Q делит отрезок AD в отношении 2:3 (считая от точки A), O – пересечение PQ и MN. Найдите отношение MO к ON.*

Любознательный читатель, наверняка, желает решить задачу самостоятельно, поэтому ее разбора здесь приведено не будет. Посмотрим на другие задачи.

Какое наименьшее значение может принять выражение $(x^2 + 2)^2 + 4x^3$*?*

В данном случае разбор задачи будет приведен позже, но перед тем как его прочитать, заинтересованный читатель, вероятно, захочет попробовать испытать свои собственные силы.

Какой остаток при делении на 5 дает число $3^{100}33^{100}333^{100} + 8^{100}88^{100}888^{100}$*?*

Можно ли узнать этот остаток? Трудолюбивый читатель наверняка захочет это проверить.

Сколько существует значений k*, при которых квадратный трехчлен* $x^2 - 2006\,x + k$ *имеет два различных целых положительных корня?*

Видимо, читатель захочет решить и эту задачу. Для этого ему понадобится теорема Виета.

Перейдем теперь к вопросу генерации условий этих четырех задач. Написание генератора для первой задачи по геометрии не составляет труда. Варьировать в условии задачи можно отношения BP/PC и AQ/QD. Генератор выглядит примерно так:

1. Выбрать x_1 и y_1, x_2, y_2 – случайные числа от 1 до 5. Назначить $BP/PC = x_1/y_1$, $AQ/QD = x_2/y_2$.

2. Вычислить ответ $MO/ON = x/y$ (для этого надо уметь решать задачу).

3. Если оказалось, что $x_1/y_1 = x_2/y_2$, перейти к шагу 1 (считаем, что если P и Q делят стороны в одинаковом отношении, то задача решается слишком просто).

4. Если оказалось, что дробь x_1/y_1 или x_2/y_2 можно сократить, перейти к шагу 1 (в условии задачи не должно быть отношений типа 2/4).

5. Если ответ $x/y = 1$, перейти к шагу 1. (в этом случае задача решается слишком просто).

6. Вывести условие и ответ.

На шагах 3–5 проверяется допустимость условия и ответов. На шаге 4, если дроби оказываются сократимыми, генерация начинается заново. Есть возможность сократить дробь самим, то есть если получилось $x_1/y_1 = 2/4$, то программа может выводить в условие данные, что $BP/PC = 1/2$. Против этого есть одно возражение. Условия задач с отношением $BP/PC = 1/2$ будут появляться чаще условий с отношением, например, $BP/PC = 4/5$. Первое отношение будет генерироваться в двух случаях: ($x_1 = 1$; $y_1 = 2$), ($x_1 = 2$; $y_1 = 4$), второе отношение будет генерироваться только в случае ($x_1 = 4$; $y_1 = 5$). В предложенном варианте генератора эти отношения уравнены в правах.

Как видно, задача про параллелограмм отлично поддается генерации, но, к сожалению, ответы в ней всегда или почти всегда получаются громоздкими. Самый частый ответ – отношение двузначных чисел. Простота генерации не влечет простоту решения задачи.

В обоих написанных ранее генераторах заметна совпадающая общая структура. Теоретически генератор любой задачи может быть написан на основе этой структуры. Приведем ее здесь:

1. Начальная инициализация генератора. (В случае генератора «Алгоритм Евклида» на этом шаге был придуман параметр D, будущий ответ).

2. Вычислить все параметры задачи. В предыдущих примерах параметрами были величины x_1, y_1, x_2, y_2, четырехзначные числа M и N. Параметры могут выбираться случайным образом (как было в предыдущих примерах) или же определяться на основе уже выбранных параметров.

3. Решить задачу, то есть вычислить ответ.

4. Перейти к шагу 2 в любом из трех следующих случаев:

a) подобрались неудачные параметры, (например, в предыдущем генераторе параметры $x_1=2$ и $y_1=4$, дающие отношение $BP/PC = 2/4$, считаются неудачными);

b) если получился слишком простой или слишком сложный ответ (ответ типа –445493/26874 следует избегать);

c) если задача решается слишком просто или слишком сложно (например, в задаче на алгоритм Евклида под сложностью решения понималось количество шагов алгоритма, сложность должна была быть от 7 до 10).

5. Вывести условие и ответ.

Подобный шаблон написания генератора универсален, и практически всегда имеет смысл писать генератор именно так. Исключения составляют особые случаи. Изредка необходимо ускорить работу генератора. Действительно, цикл, описанный в шагах 2–4, работает очень неоптимально. Другой особый случай, когда шаблон не следует использовать, стоит разобрать отдельно.

Вспомним задачу из школьного курса теории вероятности.

В 3-а классе 25 учеников. С какой вероятностью среди учеников 3-а класса двое имеют день рождения в один день?

Ответ известен (если нет, то попробуйте проверить): эта вероятность больше 50%! Какое это имеет отношение к генерации задач? Представим, что генератор способен создать 365 различных вариантов некоторой задачи и все задачи могут быть созданы с одинаковой вероятностью. Мы попросили генератор создать всего 25 задач, то есть запустили его независимо 25 раз. Как и в случае со школьниками, вероятность получить две одинаковые задачи составляет более 50%. Именно с такой вероятностью в классе из 25 человек двое будут решать одну и ту же задачу, при том что в принципе возможных разнообразных задач хватило бы на весь класс – их 365 штук.

Обобщим сказанное, если генератор способен создать N вариантов одного задания, то может быть проверено, что при генерации примерно $\sqrt{N}$ вариантов задания с вероятностью более чем 50% будет сгенерировано два одинаковых задания. Действительно, в предыдущем примере $\sqrt{365} \approx 25$. Таким образом, создание генераторов с маленьким потенциальным набором задач нежелательно. Или же такие генераторы не должны генерировать задачи случайным образом. В последнем случае описанный выше шаблон для написания генераторов применять не следует, так как он как раз генерирует задачи случайным образом.

Не будем более радовать читателя примерами генераторов и предложим ему написать их самому. Разберем вторую задачу на поиск минимума выражения $(x^2 + 2)^2 + 4x^3$. Преобразуем выражение, выделив в нем полные квадраты:

$$\begin{aligned}&(x^2+2)^2+4x^3 = \\ &= x^4+4x^2+4+4x^3 = \\ &= x^2(x^2+4x+4)+4 = \\ &= x^2(x+2)^2+4\end{aligned}$$

Выражение не может оказаться меньше 4, так как первое слагаемое является точным квадратом и, следовательно, неотрицательно. При $x = 0$ значение выражения равно как раз 4, в результате мы получаем ответ: наименьшее значение выражения равно 4. (На самой олимпиаде была предложена чуть более сложная задача на ту же идею выделения полных квадратов: требовалось найти минимум выражения $3a^2 + 4ab + 4b^2 + 4a + 3$. Разобранный вариант задачи предлагался на предварительном туре олимпиады, который проводился на несколько недель раньше).

Задача поддается генерации. Понятно, что коэффициенты 2 и 4 из условия не единственно возможные, для того чтобы задача решалась с помощью предложенного метода выделения полного квадрата. Тем не менее, коэффициенты 2 и 4 должны быть аккуратно подобраны, чтобы полный квадрат в процессе решения выделить удалось.

Для генерации предлагается параметризовать задачу следующим образом:

$$(ax^2 + b)^2 + c(x^3 + d).$$

Не составляет труда закончить писать генератор этой задачи, и вряд ли читатель откажет себе в удовольствии это сделать.

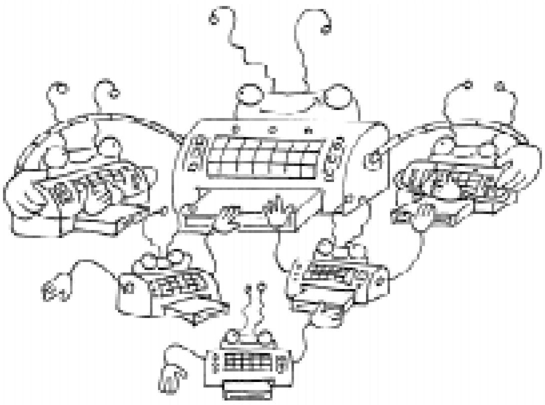
Посмотрим на примеры уже сгенерированных условий задачи:

Какое наименьшее значение может принять выражение $(9x^2 + 8)^2 + 216x^3$?

Ответ: 64

Какое наименьшее значение может принять выражение $(x^2 + 8)^2 + 8(x^3 + 3)$?

Ответ: 88



Дополнительной возможностью системы является помощь в быстром создании новых генераторов.

Какое наименьшее значение может принять выражение $(18x^2 + 25)^2 - 1080(x^3 + 1)$?

Ответ: –455

Как видно, суть задачи после генерации не изменилась, она по-прежнему решается исходным методом.

Для оставшихся двух задач олимпиады также могут быть написаны генераторы. Прокомментируем то, как это делать. В задаче про остаток по модулю 5, число 5 является частью условия и не должно меняться. Смысл числа 5 в том, что задачу в этом случае можно решать с помощью определения последней цифры $3^{100}33^{100}333^{100} + 8^{100}88^{100}888^{100}$. Генератор должен варьировать цифры 3, 8, и степень 100.

Наконец, в последней задаче про теорему Виета генератор для создания различных вариантов задачи может варьировать число 2006, знак перед числом 2006, слово «положительных» / «отрицательных» / «неположительных» / «неотрицательных».

ОБСУЖДЕНИЕ АВТОМАТИЧЕСКОЙ ГЕНЕРАЦИИ ЗАДАЧ

Автоматическая генерация задач является мощным средством защиты от списывания, ученикам выдаются разные задачи, и каждый должен сам проделать все требуемые вычисления. Генерировать задачи можно с помощью любого средства программирования, но недостаточно просто уметь генерировать задачи. Из сгенерированных условий задач следует уметь составлять контрольные, а контрольные следует уметь аккуратно размещать на бумажном листе. В статье А.В. Степанова «Система компьютерной генерации заданий по математике» (журнал «Компьютерные инструменты в образовании». 2000, № 3/4. С. 28–31) была описана система компьютерной генерации заданий по математике в своей начальной стадии разработки. Сейчас система развилась и имеет огромное количество возможностей. На рис. 2 скопирован кусочек из результата работы системы. Три задачи из Интернет Олимпиады по математике составляют контрольную, которая аккуратно отображена на листе бумаги. В возможности системы входит хранение и поиск генераторов задач, составление из них контрольных и расположение контрольных на листах бумаги. Получающиеся контрольные достаточно разрезать и использовать по назначению.

Дополнительной возможностью системы является помощь в быстром создании новых генераторов. Действительно, мы видели раньше, что, по существу, все генераторы имеют одинаковую структуру и пишутся по одному шаблону. Для создания генератора совсем не обязательно писать его

Вар. 1 1. Какое наименьшее значение может принять выражение $(x^2+8)^2-8(x^3+93)$? 2. Какой остаток при делении на 5 дает число $2^{400}22^{400}222^{400}+7^{400}77^{400}777^{400}$? 3. Сколькими способами можно присвоить значение свободному члену k квадратного трехчлена $x^2+1700x+k$, чтобы он имел два целых неположительных корня?	**Вар. 2** 1. Какое наименьшее может принять $(x^2+32)^2-16(x^3+63)$? 2. Какой остаток пр нии на 5 дает $7^{900}77^{900}777^{900}+8^{900}88^{90}$ 3. Сколькими способами м своить значение свобод ну k квадратного $x^2-7200x+k$, чтобы два целых неотрицател ня?
Вар. 4 1. Какое наименьшее значение может принять выражение $(18x^2+25)^2+1080(x^3-1)$? 2. Какой остаток при деле-	**Вар. 5** 1. Какое наименьшее может принять $(2x^2+9)^2+24(x^3-30)$? 2. Какой остаток пр

Рис. 2.

программу с нуля. Среда разработки генераторов заполняет шаблон данными, предоставляемыми программистом. Необходимы данные о тексте условия задачи и тексте ответа, о параметрах задачи и методах их вычисления, наконец, как видно из 4 шага в шаблоне, необходимы данные об ограничениях на параметры задачи и об ограничениях на ответ. Построенный по шаблону генератор является полноценным генератором и может быть использован для создания задач.

СОЗДАНИЕ ГЕНЕРАТОРОВ БЕЗ ПОМОЩИ ПРОГРАММИСТОВ

Тем не менее, процесс создания генераторов имеет ряд проблем, которые еще ждут своего решения.

Очевидно, что создание генераторов – дело рук программиста, и оно пока недоступно для обычного преподавателя. Какая бы мощная среда разработки генераторов ни была предоставлена преподавателю-непрограммисту, он, скорее всего, не сможет написать на ней нужный ему генератор. Самый первый разобранный нами пример – это пример генератора задачи на алгоритм Евклида. Представим преподавателя, создающего себе генератор задачи на алгоритм Евклида. Он сможет объяснить системе, что ответом в задаче будет НОД двух чисел a и b. НОД – это базовое понятие школьной программы, и системе создания генераторов надо его знать. Но сможет ли преподаватель объяснить системе, что «количество шагов в алгоритме Евклида» должно быть примерно одинаковым в каждой задаче? Понятие количества шагов алгоритма специфично именно для этой задачи и вряд ли оно будет введено в систему.

Аналогично, если преподавателю требуется создать генератор задач на поиск кратчайших путей в графе, то есть задач на алгоритмы Дейкстры и Флойда, или задач на поиск минимального остовного дерева, то есть на алгоритмы Прима и Краскала, то ему саму придется реализовывать эти алгоритмы. Количество возможных задач бесконечно, все возможные алгоритмы не поместить в систему заранее.

Тем не менее, сказанное выше не означает, что преподаватель не способен создавать генераторы задач сам. Достаточно ограничиться некоторым классом задач, генераторы для которых можно создавать без участия программиста. Самый простой класс таких задач – это задачи на знание формулы. Например «Длина окружности равна 42 метрам, найдите площадь круга». Для создания подобных задач преподавателю достаточно ввести в систему создания генераторов одну формулу. В данном случае формулу вычисления площади круга по длине окружности. (Может ли читатель вывести эту формулу?) Единственной формулы хватает для создания генератора.

Есть и второй случай, когда преподаватель может создавать генераторы без участия программиста. Следует ограничиться одной очень узкой и конкретной темой будущих создаваемых задач. Например, «задачи на системы счисления». В такой системе есть возможность создавать самые разные задачи с условиями типа «Переведите число a из системы счисления x в систему счисления y». «Сложите числа a и b в системе счисления x».

ХРАНЕНИЕ И ПОИСК ГЕНЕРАТОРОВ

Как было только что замечено, генераторы общего назначения может писать только программист. Работа программиста трудна и не быстра. Каждый генератор должен быть отлажен (как известно, любая программа содержит ошибки).

Из сказанного следует, что написанные генераторы следует аккуратно собирать и хранить, не стоит надеяться, что при необходимости программист напишет их заново. Но требует своего решения вопрос, как хранить написанные генераторы. Можно ли сделать удобный поиск среди нескольких сотен написанных генераторов? Еще только предстоит придумать, как именно это сделать. Читатель может попробовать свои силы в придумывании удобной системы хранения и поиска генераторов задач.

ВЫВОД ДАННЫХ В ГЕНЕРАТОРАХ

Шаблон общего генератора содержит пятый шаг «Вывести условие и ответ». Посмотрим на задачу «Известно, что $\log_5 2 = a$. Выразите через a: $\log_5 40 - \log_2 100 - \log_{10} 8$». Ответом в задаче является $3a - 1 - 2/a - 3a/(a + 1)$. Чуть изменим условие: «Известно, что $\log_5 3 = a$. Выразите через a: $-\log_5 675 + \log_3 225 + \log_{15} 5$». Ответом будет $-3a + 2/a + 1/(a + 1)$. Ответ похож на предыдущий, но все же выглядит иначе. Во втором ответе не вычитается константа, последняя дробь в числителе не имеет множителя a.

Так как задача генерируется автоматически, ответ к ней выводит генератор. Он должен проследить за аккуратным выводом ответа. Большую часть программы генератора часто составляет код, не относящийся к сочинению задачи, код относится к внешнему виду получающегося результата. Ответ и условие должны выглядеть красиво и не содержать нулевых слагаемых, не содержать умножений на единицу, возведений в первую и нулевую степень и т. п. Ответ $0a + 7b + -3c$ нельзя считать красивым, и следует позаботиться о том, чтобы вместо него выводить просто $7b - 3c$.

Некоторые создаваемые генератором ответы нельзя сразу давать преподавателю, перед этим ответ следует упростить, чтоб он был более удобен для проверки. Например, один из написанных некоторое время назад генераторов создавал ответы, подобные изображенному на рис. 3. Читатель может попробовать самостоятельно упростить это выражение, проверить, чему оно равно.

ВЫЧИСЛЕНИЯ В ЗАДАЧАХ

Здесь осталось обсудить основную проблему генераторов задач. Сложность всех приведенных выше задач состоит в вычислениях. Даже красивая задача про поиск минимума выражения $(x^2 + 2)^2 + 4x^3$ после генерации превратилась в неприятную задачу с возведениями в квадратах двухзначных чисел. Генерации в принципе поддаются только задачи, требующие вычисле-

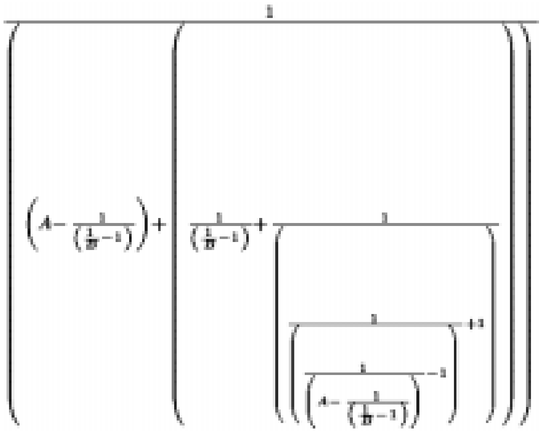


Рис. 3

ний или действий в соответствии с некоторым алгоритмом. Четыре приведенные для примера задачи из Интернет-олимпиады требуют вычислений, что затмевает их исходную красоту. Вы, читатель, конечно, решили все четыре задачи олимпиады. Попробуйте расположить задачи в порядке убывания количества решивших их участников. Сначала самую популярную задачу, потом менее популярную и т.п. Нашли самую простую задачу?

Реальное распределение результатов таково: больше всего народу решило задачу про поиск остатка числа $3^{100}33^{100}333^{100} + 8^{100}88^{100}888^{100}$ при делении на 5 – 192 человека. Вторая по сложности задача – это задача по геометрии, ее решил 131 человек. Далее идет задача про поиск минимума выражения $(x^2 + 2)^2 + 4x^3$, решили – 82 человека, и, наконец, задача на теорему Виета – 38 человек. Результаты приведены

Он должен проследить за аккуратным выводом ответа...

для 10 класса, где, как было сказано, участвовало около 700 человек. Можно назвать огромное количество факторов, повлекших за собой такой слабый результат в решении олимпиадных задач школьниками. Одним из этих факторов, к сожалению, будет именно переусложнение вычислений в задачах, оно связано с использованием генераторов для создания задач.

Чем лучше генератор, тем меньше требуется вычислений в создаваемых им задачах и тем больше требуется размышлений о том, как решается задача. Но, как было показано, не так просто создавать хороший в указанном смысле генератор.

ЗАКЛЮЧЕНИЕ

Автоматическая генерация задач является мощным средством защиты от списывания. Генераторы можно создавать с помощью любого средства программирования, причем оказывается, что все генераторы имеют схожую структуру, и поэтому генераторы могут разрабатываться с помощью средств для создания генераторов. Правда, пока что доступно только одно такое средство. Существуют проблемы автоматического генерирования задач, которые еще ждут своего решения. Это разработка удобной системы хранения и поиска генераторов задач, это увеличение доступности разработки генераторов для преподавателей-непрограммистов. Также необходимо научиться тратить как можно меньше усилий на вывод условий и ответов. Действительно, генератор не должен заботиться о том, чтоб упрощать математические выражения в условиях и ответах. Система Mathematica, например, умеет упрощать любое математическое выражение, и было бы неплохо уметь делать что-либо подобное.

Необходимо научиться разрабатывать генераторы для невычислительных задач. Пока что все генерируемые задачи для своего решения требуют серьезных вычислений, и ничего другого генерировать пока не получается.

Ну и, наконец, спасибо тебе, читатель, что дочитал эту статью до конца.

Посов Илья Александрович,
аспирант математико-механического факультета СПбГУ.