Неполные и неточные задачи в обучении информатике

Непейвода Н. Н.

февраль 2025

Введение

Постановка проблемы 1

В реальности программист и информатик будет почти всегда (кроме работы в нескольких уникальных фирмах) получать в лучшем случае неполное, а часто ошибочное, «техническое задание». Худший случай, когда оно имеет при этом внешнюю форму полного и точного и оформлено по всем ГОСТам.

Постановка проблемы 2

Ситуацию резко изменило также появление ГПТ. В частности, в МГТУ оказались студенты-отличники, практически не умеющие программировать к третьему курсу (если и умели, то разучились) и решавшие все задачи при помощи ГПТ, пока не наткнулись на курс, где задачи были действительно нестандартные, а преподаватель оценивал, насколько человек может понимать и модифицировать *свою?* программу.

Решение до появления проблемы

В практике обучения в УдГУ и на ФИТ НГУ применялись неточные и неверные постановки задач, чтобы отрабатывать навыки критического отношения к формулировкам и их уточнения формальными и неформальными методами. Это было применено также в олимпиадах (Всесибирской и Удмуртской). Поскольку это практиковалось достаточно длительное время, был накоплен некоторый опыт. Его систематическое изложение требует большой статьи, поэтому в тезисах ограничимся несколькими примерами.

Игровые задачи

Крестики-нолики

На Всесибирской олимпиаде предлагалась следующая задача. Упрости правила гомоку (=крестики-нолики) отменив все запреты. Можно выигрывать больше чем пятью шашками в ряд. После этого представляется с теоретической точки зрения вероятным, что можно построить программу вообще без расчёта ходов, генерируя поле оценок ценности клеток и пересчитывая его методом динамического программирования после каждого хода. Поэтому была дана игровая задача со следующими условиями.

Формат данных

Программа выдавала массив 19×19 целых чисел, и программа жюри случайно выбирала максимальное значение оценки. Время на ход, начиная со второго, жестоко ограничивалось: 0.01 сек. во избежание расчёта вариантов.

Сюрприз

На коленке жюри написало свою игровую пробную программу и первые две игры были у студенческой с пробной. В дальнейшем программы участников играли между собой по две партии, за выигрыш у того, кто проиграл пробе две партии, давалось одно очко, кто выиграл одну - два, выиграл обе - четыре. Заранее это не объявлялось, чтобы шибко хитрый не ухитрился вставить в свою программу проигрыш первых двух партий.

Шибко умные и шибко хитрые

Тем не менее команда ЛИТМО ухитрилась вставить в свою программу элементарный форсированный расчёт. Жюри посмеялось, подискутировало, и приняло соломоново решение: оштрафовать их на $n - 1$ балл, где n расстояние от ближайшего соперника, и сохранить им почти честно завоёванное первое место.

Одна из слабейших программ получила большую премию за неформатную выдачу. Вместе с очередным полем она напечатала, что поле заполнено, ход невозможен. Когда эту строку удалили и запустили программу жюри, та сдохла. За найденную ошибку жюри был приз. Но как нужно было плохо играть и ей, и противнику, чтобы заполнить поле идиотскими ходами.

Полная открытость

Жюри анализировало код программ участников, и выложило для всеобщего анализа все программы и все логи партий. Об этом участники заранее были предупреждены. И поэтому никто, кроме ЛИТМО, не вставлял просмотр вариантов, а ЛИТМОшники вставили его в настолько обфусцированном виде, что разобралось лишь жюри.

Одно из теоретических следствий

Контест по обфусцированным программам на C ГПТ теоретически не сможет пройти, поскольку эти программы должны вдобавок быть корректными и эффективными.

Индуктивные задачи и автотестирование

Симплегады

В мифе про аргонавтов Дарданеллы блокировали две скалы, периодически сталкивавшиеся и уничтожавшие всё, что между ними. Они застыли на месте после того, как между ними успела пролететь ласточка.

В задаче два корабля с разными скоростями стоят перед входом в пролив и наблюдают за скалами, а затем идут на прорыв. Задатся премия за проход. Она уменьшается на 1 за каждый ход ожидания и на 10 за потерю одного из кораблей. Симплегадами управляет программа, в каждом тесте своя.

Управление

Как исходные данные программа студента получала скорости обоих кораблей (число ходов, необходимое данному кораблю для прохода пролива) и общее количество очков за прохождение (оно же ограничение времени ожидания). Далее при каждом обращении к управляющей симплегадами программе: логическое значение, открыты ли на этом шаге скалы. В ответ она выдавала числа 0, 1 или 2: стоим на месте либо стартует один из кораблей.

Был сделан примитивный визуализатор, показывающий ход исполнения программы, перед началом решения показывалась пара проходов (удачный и неудачный).

Тесты жюри и свои

Труднейший тест имел оценку 300. Интервалы открытости росли от 1 до 10, затем уменьшались до 1 и опять росли. Скорости кораблей были 8 и 9. Кроме того, каждый участник имел право написать собственную программу длиной тела процедуры не более 12 операторов и 200 символов, снабжённый стартовыми данными. Если его программа проходила собственный тест, то тест включался в общий пакет. Выше всего (100 очков) оценивался тест, который проходила половина участников, затем оценки падали и за тест, который проходили все либо лишь сам автор - поощрительные 10 очков.

Коварный тест жюри

Тесты написали все участники. Тесты жюри были детерминированными, а участников часто рэндомные. Победил чисто рэндомный тест, который с вероятностью половина открывал ворота на 1 шаг либо на всё время прохождения.
В единственном рэндомном тесте жюри интервалы открытия менялись случайно, но почти всегда меньше скорости, пока не пожертвован кораблю, после чего ворота оставались открытыми 20 ходоа (для достаточно образованных участников косплей Сциллы и Харибды).

Переписать программу

Гонки

На первом этапе было коллективное соревнование. Поле 100×100 задавало кольцевую трассу гонок. Было четыре поля, на одном стартовали группами по 5 болидов, на другом по 4, на третьем по 2, на четвёртом, которое заранее не показывалось, могли по 10, на самом деле участников просто поделили на 2 группы, так как команд было меньше 20. Программы участников управляли ускорением болидов, которое могло быть от -2 до 2 по обеим координатам. На каждом ходу программам давалось всё поле. Ходы всех болидов делались одновременно. В первый день тот, кто врезался в чужой болид, останавливался и передавал ему весь остаток своей скорости, если протараненный врезался в стену, он выбывал из гонки.

Разбор программ

После конца тура жюри вызвало участников для разбора их программ и начало формально критиковать их оформление, добиваясь структурированности текста и ясных комментариев. Каждое возражение каралось снятием очка, так что создавалась атмосфера разбора на фирме. Сами программы при этом не критиковались и не изменялись.
Естественно, в первый день выигрывали самые тупо агрессивные.

Сюрприз на второй день

На второй день командам по жребию давались чужие программы и говорилась вводная. Теперь таран карается снятием с соревнований, а протараненный останавливается на ход. Переделайте чужую программу так, чтобы она работала в этих условиях. Оцениваться будет также соответствие текста старой и новой программ, так что изменяйте поменьше. Запускалась только самая широкая трасса, но это заранее не говорилось. Соответствие оценивалось чисто формально, процентом совпадения текстов с точностью до перестановки операторов, так что авторы многословных комментариев давали фору своим соперникам. Полученные баллы снижались на процент несоответствия.

Неполные условия

Сделай хорошо сам не знаю что

Формулировка давалась в духе невежественного самоуверенного начальника, в размытой форме. Единственное, что строго определялось — формат данных и интерфейсов. На первом этапе участники писали программы, базируясь на «техзадании» и четырёх достаточно тривиальных тестах, а затем могли доработать его, предложив свои шесть тестов. Затем велось официальное тестирование, каждому из участников давались полные списки тестов и результаты прогонки его программы. Они дорабатывали свои программы и было второе окончательное тестирование.

Пример связанной пары задач

Бутылка и бублик 1

Первый тур. Гомеоморфизм двух фигур означает возможность непрерывно и взаимно-однозначно преобразовать каждую из них в другую (таком образом, здесь не может быть разрывов и отождествлений). Например, треугольник гомеоморфен квадрату и кругу, но не кольцу либо сфере.
Двумерная поверхность (многообразие) задана следующим образом. она разбита на куски, гомеоморфные замкнутым треугольникам (симплексы). у каждого симплекса некоторые из сторон могут быть склеены со стороной другого либо того же симплекса. каждая из сторон может участвовать лишь в одной склейке.

Бутылка и бублик 2

Склейка может производиться таким образом, что склеиваемые стороны ориентированы в одну и ту же сторону либо противоположно. Направления сторон в каждом симплексе: первая от третьей ко второй, вторая от первой к третьей, третья от второй к первой. Если для склейки требуется отождествить другие вершины, за исключением отождествляемых непосредственно при склеивании сторон, задание многообразия считается ошибочным.
Число симплексов в многообразии не более 100.
многообразие задается записью следующего вида.
<число симплексов> <число отождествлений>
<номер симплекса> <номер стороны> <номер симплекса>
<номер стороны> <ориентация склейки>

Бутылка и бублик 3

Пример
2 3
1 1 2 2 +
1 2 2 3 +
1 3 2 1 +
Это означает, что в многообразии два симплекса, первая сторона первого склеена со второй стороной второго в том же направлении, вторая сторона первого с третьей стороной второго, а третья с первой стороной второго. Легко видеть, что данное многообразие гомеоморфно сфере. ориентация задается символами ’+’ или ’-’.

Бутылка и бублик 4

Входной файл input.txt содержит последовательно заданные два многообразия. Нужно вывести на экран три текстовых строчки. В первой говорится, гомеоморфны ли они, во второй какому из стандартных многообразий эквивалентно первое многообразие, в третьей какому эквивалентно второе. Если не удается установить гомеоморфность чему- то стандартному, то вторая (третья) строчка должна содержать символы ’???’. Если заданное многообразие просто невозможно, то это нужно записать в качестве текстовой строчки.

Бутылка и бублик 4

Программа должна распознавать следующие стандартные многообразия.
круг
сфера
кольцо
кольцо мебиуса
тор
бутылка клейна
проективная плоскость
Не запрещается в некоторых случаях распознавать и другие многообразия, вместо того, чтобы писать ???.

Бутылка и бублик Пример

Вывод не обязательно должен быть в точности такой и этот не обязательно правильный.

2 3
1 1 2 2 +
1 2 2 3 +
1 3 2 1 +
1 0

вывод
негомеоморфны
сфера
круг

Бутылка и бублик Пример

2 3
1 1 2 2 -
1 2 2 3 +
1 3 2 1 +
2 1
1 1 2 3 -
негомеоморфны
невозможно
круг

Бутылка и бублик Примечание

Невозможность означает некорректность определения данного многообразия вообще, а не невозможность вложить его без самопересечений в наше трехмерное пространство (например, бутылку клейна в него не вложишь без самопересечений). Также и возможность преобразовать многообразия друг в друга не означает существование непрерывной деформации нашего пространства, их преобразующей. Узлов мы не рассматриваем. Для желающих сообщаем, что для вложения без самопересечений любых двумерных многообразий достаточно четырехмерного пространства.

Тур 2. Гонки на бутылке

Вы можете использовать идеи либо фрагменты из нее для решения следующей задачи. в файле bottle.txt находится определение многообразия в том же виде, что и на первом туре: через склейки сторон треугольников, например 4 4
1 2 2 1 -
1 3 4 1 -
2 3 3 3 -
3 2 4 2 -
В первой строке количество треугольников и количество склеек.

Гонки на бутылке 2

В файле init.txt находится начальная позиция вашего игрока (охотника) и жертвы. Позиция задается следующей строкой:
3 + 15 - 21
Первое число — номер треугольника, в котором находитесь вы, далее идет его ориентация (если ’+’, то нумерация его сторон идет для вас против часовой стрелки, т. е. в положительном направлении). Далее те же данные для жертвы. Предпоследнее число ваша скорость, последнее скорость жертвы.
Внимание! скорость жертвы может быть 0.

Гонки на бутылке 3

Ваша задача — за минимальное число шагов нагнать жертву. Жертва поймана, если вы оказались на одной и той же поверхности одного и того же треугольника. если вы выяснили, что многообразие несвязно или вы на разных сторонах двусторонней поверхности (и только в этом случае) вы первым ходом выдаете в файл move.txt строку "Surrender". если же скорость ваша недостаточна для поимки, вы все равно должны пытаться, надеясь на тупость жертвы. На поимку отводится максимум 100 ходов.

Гонки на бутылке 4

Ваше движение состоит в переходах из треугольника в треугольник через общую вершину. Движение жертвы — через общую сторону. вы можете не использовать все разрешенные вам ходы и даже стоять на месте, жертва использует все ходы обязательно.
Ваши данные вы можете хранить в файле hunter.txt. Формат данного файла и его содержимое остается на ваше усмотрение. Перед вашей работой жюри не обязано очистить данный файл от мусора, оставленного вами либо другими.

Гонки на бутылке 5

Ваша программа компилируется как файл hunter.exe и будет вызываться для очередного хода другой программой. При ее втором и последующем вызовах в файле move.txt находится номер и ориентация треугольника, в котором находится жертва. На поимку вызывающая программа среагирует сама.

Гонки на бутылке 6

В файл move.txt вы выдаете после очередного вызова последовательность строк вида (например):
2
2 -
4 +
где первое число — число ваших ходов (не больше вашей скорости, может быть нулем), следующие строки — последовательные треугольники, проходимые вами, если в первой строке не ноль.
Внимание! Задание многообразия удовлетворяет следующим ограничениям. никакие три стороны не склеиваются друг с другом. Не склеиваются две стороны одного и того же треугольника. Никакие два треугольника не склеиваются больше, чем одной стороной.

Гонки на бутылке 7

Дополнительные условия
Для получения премиальных очков вы имеете право представить программу eluder.exe, которая играет за жертву. Файл, в котором вы храните вспомогательную информацию, eluder.txt. Забота о контроле сохранности файлов hunter.txt или eluder.txt лежит на вас самих. Если вы поймали противника на изменении вашего файла, вам засчитывается максимальное число очков за данный тест. В этом случае вы должны выдать в файл move.txt осмысленное сообщение о жульничестве, желательно с информацией, которая поможет жюри доказать нечестную игру противника.

Гонки на бутылке 8

Программа жюри нечестностей выявлять не будет, но будет контролировать корректность ваших ходов (и вашего противника, если он бьется за дополнительные очки). За ложное обвинение вы получаете штраф.
Использование вами фрагментов из вашей программы первого тура никак не регламентируется и близость вашей программы к программе первого тура никак не оценивается.

Заключение

Почему было возможно

Непосредственная поддержка фирм (в Удмуртии Марк-ИТТ, в Новосибиске Новософт) и, как следствие, максимальная независимость.
Участие таких сильных организаций, как ЛИТМО.
Поддержкой от них был сам факт посылки своих команд на совершенно неофициальные олимпиады.

Почему пропало

Прежде всего, из-за формализации требований к олимпиадам в связи с превращением их в инструмент параллельных вступительных экзаменов в ведущие вузы.
Такие соревнования в неформальном духе всегда не нравились чиновникам.

Сборник

Часть задач нестандартных олимпиад были включены в методическое пособие. В этот момент нестандартные олимпиады как раз были убиты стандартными.
А. Е. Анисимов, В. В. Пупышев
Сборник задач по основаниям программирования.
УдГУ 2005
Его файл FPT.PDF я скинул на компьютер конференции

Вопросы
&
Открыт для дискуссии в кулуарах