ИНФОРМАЦИОННОЕ ПРОСТРАНСТВО

Романовский Иосиф Владимирович,
Черкасова Полина Геннадьевна

ПАКЕТ ДЕМОНСТРАЦИОННЫХ ПРОГРАММ ПО КУРСУ ДИСКРЕТНОГО АНАЛИЗА «DADemo»

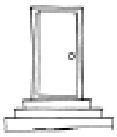
КРАТКИЙ ОБЗОР

Сейчас в информатике и родственных областях имеется потребность вводного курса дискретной математики. Обычно он включает множество интересных алгоритмов, основные возможности которых становятся понятными только после многочисленных опытов. Кроме этого, интересные алгоритмы, исключенные из программы, могли бы заинтересовать студентов, которые хотели бы расширять данную область знаний.

Учитывая все это, мы несколько лет назад начали разрабатывать коллекцию демонстрационных программ в различных областях дискретной математики. Помощь наших студентов была очень кстати. Сейчас мы располагаем некоторым количеством таких программ. Среди них имеются как очень простые демонстрационные программы, так и модели весьма сложных алгоритмов.

С целью объединить все эти программы в одно целое мы разработали систему, состоящую из очень простой оболочки (DADemo) и вспомогательных инструментов. DADemo ведает коллекцией программ, позволяет выбрать одну из них и запустить. Каждая программа реализована в виде экспортируемой функции DLL библиотеки (Dynamic Link Lib-rary). Два года наши студенты создавали свои модули исключительно в таком формате.

Вся информация о DLL библиотеках и вызываемых функциях заключена в файле DA.ini. Этот файл определяет всю конфигурацию системы для данного компьютера.

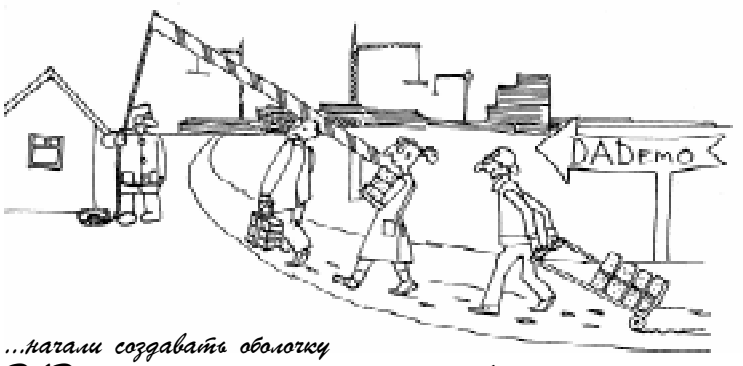


ВВЕДЕНИЕ

Многие студенты, если они учатся программировать, пишут маленькие программки, которые визуализируют работу тех или иных изучаемых алгоритмов. Некоторые делают это по заданию преподавателя, некоторые – по собственному вдохновению. Часть таких программ получаются очень удачными. Но, как правило, эти произведения искусства пропадают без толку. Объединить их в какой-нибудь каталог сложно, потому что эти программки имеют различную архитектуру – одни пишут и читают данные из различных непонятных мест жесткого диска, другие самыми разными способами используют системный реестр. Одни программы работают только с конкретной операционной системой, а другие используют специфические драйверы. Очень жаль, что хорошая работа пропадает зря.

И при этом хочется иметь коллекцию программ для иллюстрирования курса дискретного анализа.

Все эти соображения привели к тому, что несколько лет назад мы начали создавать оболочку под названием DADemo, которая могла бы объединить в одно целое множество демонстрационных программ, написанных разными людьми в



...начали создавать оболочку DADemo, которая могла бы объединить в одно целое множество демонстрационных программ, написанных разными людьми в разное время и на различных языках рограммирования.

разное время и на различных языках программирования.

ЖИЗНЕННЫЙ ЦИКЛ

Жизнь такой оболочки представляется таким образом. Сначала мы придумали стандарт для демонстрационных программ, то есть перечислили требования, которым должна удовлетворять демонстрационная программа, для того чтобы ее можно было поместить в коллекцию. Затем написали оболочку, которая может работать с программами, удовлетворяющими этому стандарту. Затем создали несколько демонстрационных программ и показали все это студентам, которые изучают дискретный анализ.

В результате, во-первых, студентам стал понятней изучаемый курс. А во-вторых, некоторые студенты вдохновились и захотели тоже написать что-нибудь для нашей коллекции демонстрационных программ. Мы показали им список требований к программам, и они приступили к работе. Ряд студентов написали демонстрационные программы, которые нам понравились. Мы делали мелкие замечания. Студенты вносили исправления. После этого мы помещали эти программы в нашу коллекцию DADemo.

На следующий год новые студенты смотрели эти демонстрационные программы, вдохновлялись и, таким образом, наша коллекция с каждым годом пополнялась новыми работами.

Правда, у студенческих работ есть один недостаток: они, в большинстве своем, не направлены на то, чтобы кого-то обучить новому алгоритму. Эти работы говорят: «Смотрите, как красиво работает этот алгоритм». А иногда: «Смотрите, как хорошо я умею программировать». И человеку, который знает этот алгоритм, действительно нравится смотреть такие демонстрационные программы. Но если пользователь не знает алгоритма, то эта программка не поможет его изучить.

Большинство замечаний и, как следствие, исправлений в студенческих работах касаются именно этого момента. Да что говорить про студентов, – мы и сами этим иногда страдаем.

АРХИТЕКТУРА И ИНТЕРФЕЙС

Оболочка программы устроена таким образом. Запускается программа DADemo, и пользователь видит список разделов дискретного анализа. В разделах перечислены алгоритмы, понятия, механизмы, соответствующие данной теме. Пользователь может выбрать один из пунктов и запустить демонстрационную программу.

Программа DADemo поддерживает два языка – русский и английский. Переключение языков производится нажатием кнопки в главном окне. Выбранная демонстрационная программа запускается на том же языке, который выбран в главном окне.

Демонстрационные программы содержатся внутри dll библиотек, а вся конфигурация оболочки содержится в файле DA.ini. Конфигурация включает в себя список разделов, названия dll библиотек, имена процедур в этих библиотеках, которые следует вызвать для запуска демонстрационной программы, названия разделов и пунктов на русском и английском языках (см. рисунок 1).

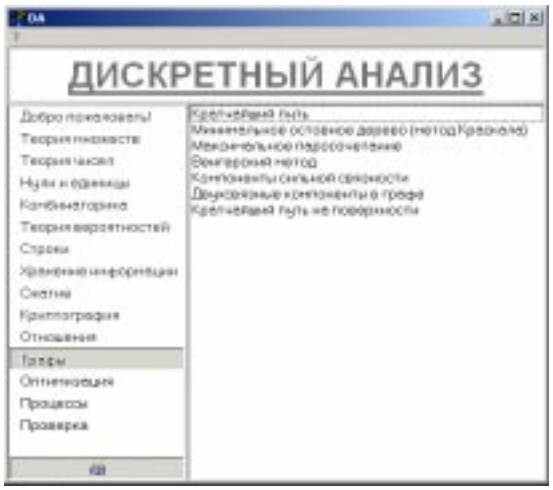
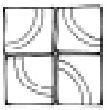


Рисунок 1

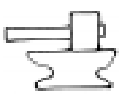


КОМПОНОВКА

Для того, чтобы добавить в коллекцию еще одну демонстрационную программу, нужно в любой среде программирования на любом языке написать dll библиотеку, в которой реализуется эта программа. Библиотека должна экспортировать процедуру, которая запускает демонстрационную программу. Этой процедуре передается один логический параметр IsRussian. В соответствии с этим параметром программа должна запускаться либо на русском, либо на английском языке. Можно создать упрощенный вариант библиотеки, который игнорирует этот параметр и работает только с английским языком. Библиотека не должна ничего писать на жесткий диск компьютера.

После этого следует поместить информацию о библиотеке в файл DA.ini в нужный раздел или создать для нее новый раздел.

Теперь в списке появится еще один пункт с новой демонстрационной программой. Выбрав этот пункт, пользователь запустит созданную библиотеку.



ТЕКУЩИЕ РАЗРАБОТКИ

В следующей версии пакета мы планируем создать так называемые хранилища данных.

Дело в том, что, как известно, любому алгоритму для работы нужны исходные данные. Для алгоритмов сортировки это набор чисел или других сортируемых объектов. Для алгоритмов сжатия это сжимаемая информация, например, текст или картинка. Для алгоритма поиска подстроки это опять текст и искомая строка. А для алгоритма поиска опорного дерева в графе это, конечно, граф.

Было бы интересно запускать разные алгоритмы с одними и теми же данными и сравнивать результаты. Для этого нужно, чтобы данные хранились в одном месте, доступном всем демонстрационным программам, – хранилище данных. Тогда при загрузке, например, программа сортировки спрашивала бы у этого хранилища список всех числовых последовательностей и предлагала бы пользователю выбрать одну из них для проведения сортировки.

Планируется также предоставлять пользователю в каждой такой демонстрационной программе кнопку, по которой вызывается редактор хранилища данных. Тогда пользователь мог бы добавлять в хранилище свои наборы данных и запускать на них работу алгоритмов.

Было бы также интересно заносить в общую таблицу статистику по алгоритмам. Каждый алгоритм во время работы с конкретным набором данных считает количество выполненных сравнений и перемещений или просто количество шагов. После этого он помещает в хранилище данных эти числа.

Для некоторых алгоритмов можно оценивать и качество результата. Например, коэффициент сжатия информации или высоту построенного дерева.

Таким образом, пользователь может увидеть, например, для нескольких алгоритмов сжатия скорость запаковки, скорость распаковки и коэффициент сжатия одного и того же текста.

Сейчас мы разрабатываем архитектуру хранилища данных. Мы стараемся, чтобы она была максимально проста. Чтобы человек, который собрался написать демонстрационную программу с использованием

хранилища данных, не тратил бы много времени на изучение его интерфейса.

ПРИМЕРЫ

Здесь приведено несколько примеров демонстрационных программ. Мы поделили их условно на две категории – игрушечные и серьезные. Игрушечные – это такие, которые не требуют от пользователя глубоких знаний, которые приятно посмотреть, с которыми можно поиграть. Они помогают пользователям перейти «на ты» с изучаемыми объектами.

Серьезные программы позволяют провести исследование этих объектов, поставить интересный эксперимент, наглядно увидеть и лучше понять работу алгоритма.

ПРИМЕР 1. РАЗБИЕНИЯ МНОЖЕСТВ

Автор: Ю. Солдак.
Под редакцией Полины Черкасовой.
Игрушечная программа.

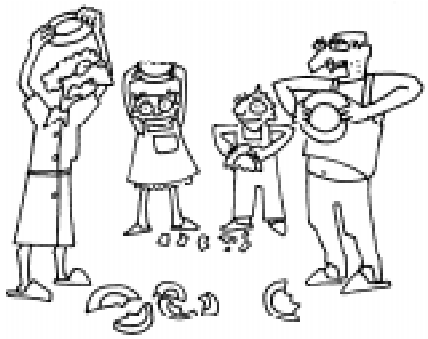
Краткий обзор

Эта программа наглядно показывает – что такое разбиения множеств и их произведение.

Множество здесь представлено в виде тарелки, а разбиение множества – в виде разбитой тарелки. А когда разбитые тарелки «накладываются» друг на друга, мы получаем произведение разбиений.

Интерфейс

При запуске программы пользователь видит MDI окно со стопкой тарелок. Кликнув на одну из них мышкой,



...что такое разбиения множеств...

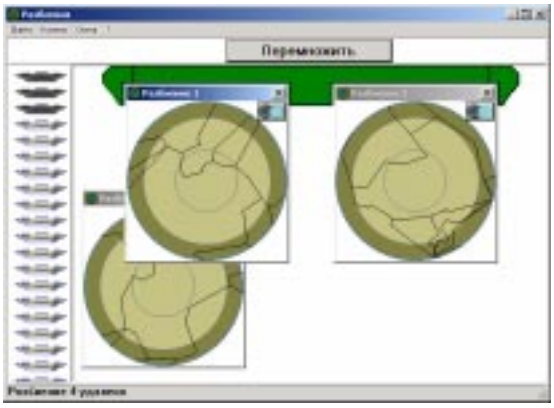


Рисунок 2

он может «взять» эту тарелку, то есть увидеть ее в виде SDI окошка.

Указатель мыши над появившейся тарелкой сделан в виде молотка. Пользователь может стукнуть этим молотком по тарелке, и тарелка разобьется, – мы получим разбиение множества. Пользователь может стукнуть по одному из кусков тарелки еще раз, – кусок разобьется, и получится более мелкое разбиение.

Если пользователь нажмет кнопку рядом с тарелкой, то появится еще одна тарелка, разбитая точно так же, – копия разбиения.

Для перемножения разбиений в окошке имеется «Перемножатель» – устройство с двумя зажимами и кнопкой «Перемножить». Если в каждый зажим вставить по разбитой тарелке и нажать кнопку, то изображения тарелок наложатся друг на друга, и получится третья разбитая тарелка, представляющая произведение первых двух (рисунок 2).

ПРИМЕР 2. РЕШЕТО ЭРАТОСФЕНА

Автор: П. Черкасова.
Игрушечная программа.

Краткий обзор

Программа наглядно показывает механизм работы решета Эратосфена для чисел до 65.

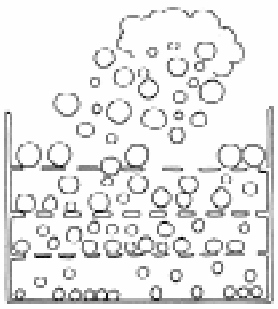
Интерфейс

В окошке программы нарисована горизонтальная линия – полочка,

вдоль которой выстроены числа от 2 до 65. Алгоритму требуется разделить их на простые и составные.

Вдоль полочки рисуется кривая, которая отмеряет на ней равные отрезки. Там, где кривая касается полочки, в ней образуется отверстие. Затем в отверстия высыпаются числа. Эти числа кратны длине отмеряемого отрезка.

В итоге на полочке остаются только простые числа (рисунок 3).



...механизм работы решета Эратосфена для чисел до 65.

ПРИМЕР 3. РЕШЕТО ЭРАТОСФЕНА ДЛЯ ДИАПАЗОНА ДО 10000

Автор: И. Романовский.
Серьезная программа.

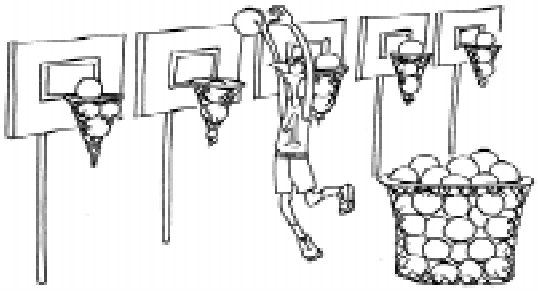
Краткий обзор

Программа показывает рисунок отсева простых чисел в диапазоне от 2 до 10000.

Интерфейс

В окошке программы нарисован квадрат 100×100. Каждая клеточка представляет число. Все числа серого цвета, только 0 белого и 1 – желтого. Рядом с квадратом имеется кнопка «Простое 2». После нажатия на кнопку все числа, кратные двум, становятся черными, сама двойка – красной, тройка – сиреневой, а на кнопке название меняется на «Простое 3».

Распределение цветов таково: красный – простые числа, по которым уже произошел отсев; сиреневый – простые числа, по которым еще не было отсева; серый – про число еще ничего неизвестно; черный – составное число (рисунок 4).



...рисунок отсева простых чисел в диапазоне от 2 до 10000.

ПРИМЕР 4. КЛАССИФИКАЦИЯ ПРЕДИКАТОВ.

Авторы: П. Холькин, Д. Филатов.
Под редакцией Полины Черкасовой.
Серьезная программа.

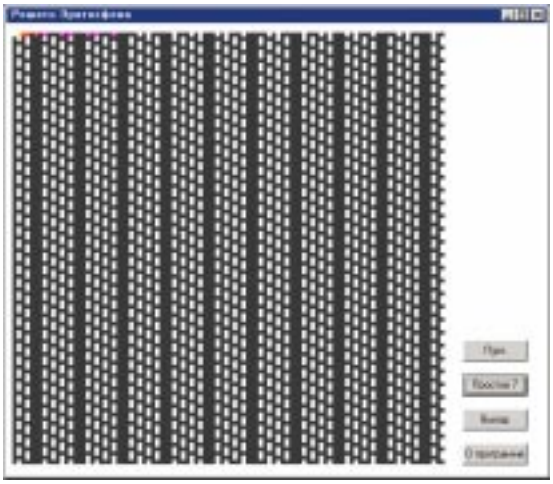


Рисунок 4

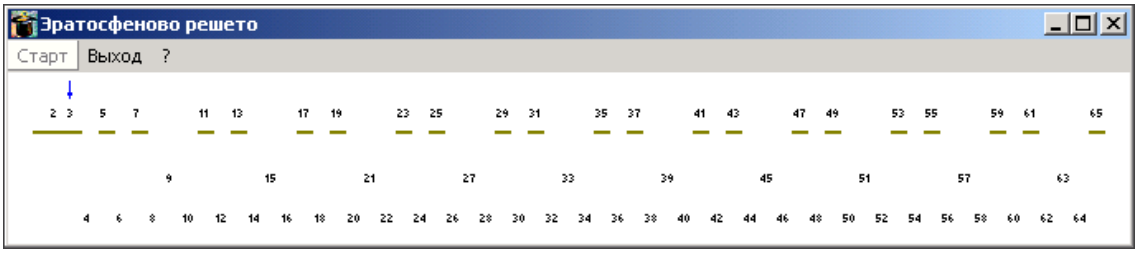


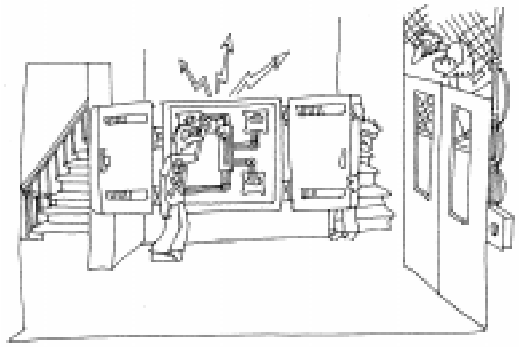
Рисунок 3

Краткий обзор

Пользователю предоставляется возможность задать отношение внутри некоторого множества, после чего происходит проверка – является ли это отношение отношением эквивалентности, отношением частичного порядка. Пользователь также может построить транзитивное замыкание.

Интерфейс

В окошке программы нарисована матрица, строки и столбцы которой проиндексированы латинскими буквами. Некоторые клетки пусты, в некоторых стоит единица. Эта матрица задает отношение на множестве, элементы которого обозначены латинскими буквами. Пользователь может менять размерность матрицы (до 7), а также убирать и ставить единицу в клетках щелчком мыши.



...транзитивное замыкание

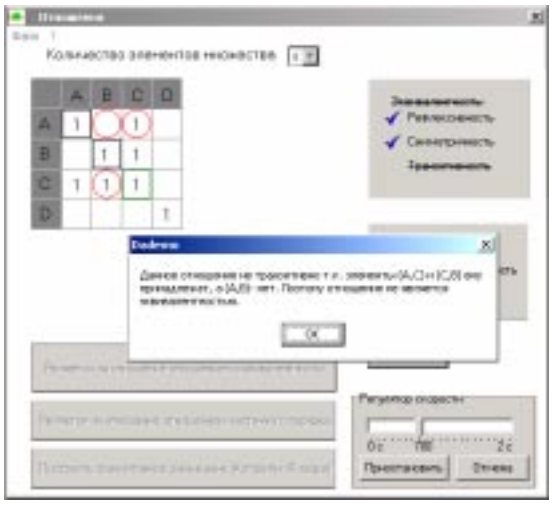


Рисунок 5

После задания отношения, пользователь может нажать одну из трех кнопок: «Является ли отношение отношением эквивалентности», «Является ли отношение отношением частичного порядка», «Построить транзитивное замыкание».

Во время исследования отношения проверяемые ячейки помечаются графически. Как только программа находит ячейки, не удовлетворяющие требуемому свойству, выводится сообщение с обоснованием (рисунок 5).

ПРИМЕР 5. ЦЕПИ МАРКОВА

Авторы: М. Плискин, Е. Хорьков.

Под редакцией Полины Черкасовой и Иосифа Владимировича Романовского.

Серьезная программа.

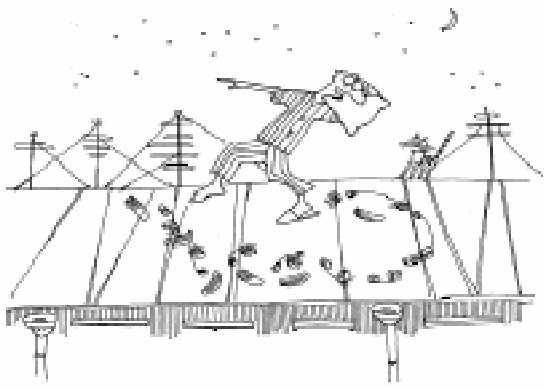
Краткий обзор

Пользователю предоставляется возможность задать матрицу переходов в Марковской цепи.

Далее запускается процесс перехода из состояния в состояние в соответствии с заданной вероятностной матрицей. Траектория состояния записывается.

Интерфейс

В окошке программы изображена матрица переходов. В ячейке (M, N) записана вероятность перехода системы из состояния M в состояние N. Нетрудно



Пользователю предоставляется возможность задать матрицу переходов в Марковской цепи

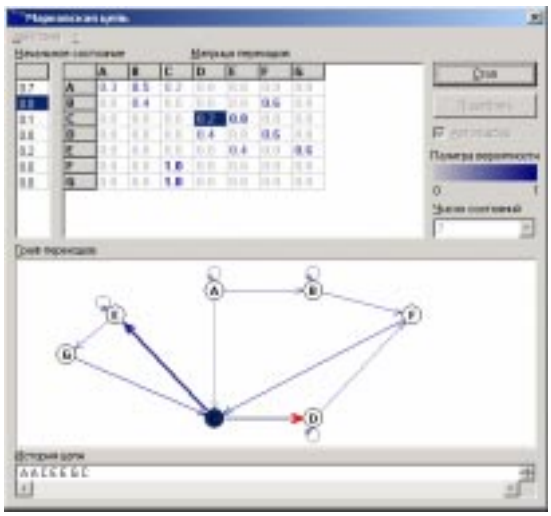


Рисунок 6

догадаться, что сумма чисел в строчках равна единице.

Также имеется столбец начального состояния: в ячейке N записана вероятность того, что начальным будет состояние N.

Пользователь может менять вероятности и размерность столбца и матрицы.

На картинке рядом вероятности переходов изображаются графически в виде графа. Чем меньше вероятность перехода, тем бледнее стрелка между вершинами.

По кнопке «Запуск» процесс запускается и графически отображается на той же картинке. Под картинкой записывается траектория состояний (рисунок 6).

Примечание: полная версия программы «DADemo» находится на прилагаемом к журналу диске.

Романовский Иосиф Владимирович,
доктор физ.-мат. наук,
профессор СПбГУ.

Черкасова Полина Геннадьевна,
программист-разработчик,
компания «Avicode» (США).