Посов Илья Александрович

COMPUTER SCIENCE КЛУБ В САНКТ-ПЕТЕРБУРГЕ

Представьте открытые для всех желающих курсы и лекции по Computer Science, которые регулярно проходят в центре Санкт-Петербурга. С сентября 2007 года петербургские и иностранные преподаватели каждую неделю в течение учебного года собираются в Петербургском отделении Математического института им. В.А. Стеклова (ПОМИ) и проводят занятия для всех, кто желает углубить свои знания в различных областях информатики.

Выпускник ИТМО Антон Лиходедов несколько лет назад решил сделать что-либо полезное для Computer Science в Петербурге и обратился за советом к Эдуарду Алексеевичу Гиршу, старшему научному сотруднику ПОМИ. В результате год назад открылся Computer Science клуб, о котором написана эта статья. Единственным спонсором клуба на данный момент является Антон Лиходедов. В своем журнале об истории создания клуба он написал подробнее: likh.livejournal.com/96044.html.

Курсы в клубе имеют разные направления, вот некоторые из тех, которые были прочитаны в прошлых семестрах.

- **Сложностная криптография** (*Эдуард Алексеевич Гирш*, ПОМИ РАН);
- **Введение в моделирование и верификацию программных и аппаратных систем** (*Борис Юрьевич Конев*, University of Liverpool);
- ***Networking*** (*Antonio Carzaniga*, University of Lugano;
- ***Graph Algorithms*** (*Andrew Goldberg*, Microsoft Research);
- **Эффективные алгоритмы** (*Александр Сергеевич Куликов*, ПОМИ РАН; об этом курсе далее мы расскажем подробно).

Полный список курсов можно найти на сайте клуба, его адрес http://logic.pdmi.ras.ru/~infclub. На сайте вы также найдете информацию о том, где проходят занятия, расписание занятий, информацию о преподавателях, и конечно, новости, в которых появляются объявления о деятельности клуба. Ко всем проведенным докладам на сайте выложены слайды и в большинстве случаев видео. На прилагающемся к журналу диске вы найдете слайды и видео, относящиеся к одному из курсов, упомянутых выше.

Для понимания практически всего, что рассказывается в клубе, достаточно минимальных начальных знаний в теоретической информатике. Для тех, кому начальных знаний все же не хватает, был проведен курс «Ликвидация безграмотности», который читал аспирант ПОМИ РАН Дмитрий Михайлович Ицыксон.

Рис. 1. Лекция Э.А. Гирша в ПОМИ РАН

Обычно занятия проходят во время выходных, чаще всего в воскресенье. Но статистика показывает, что студенты готовы тратить свое свободное время на посещение курсов, среднее количество слушателей на лекции за прошедшие два семестра – 40 человек. Небольшая статистика по участникам: за два семестра клуб посетило 217 человек, из них 94 – студенты СПбГУ, 75 – студенты СПбГУ ИТМО, 39 – студенты СПбГПУ, СПбГЭТУ «ЛЭТИ», СПбГУАП, СПбГУТ, 9 – не студенты вузов.

После окончания курсов желающие могут сдать экзамены, студентам СПбГУ и СПбИТМО оценка за них может быть зачтена формально, а в будущем планируется расширить ряд вузов, студенты которых смогут получать официальную отчетность по экзаменам.

Необходимо повторить, что занятия в клубе абсолютно свободны и открыты для всех желающих. Чтобы послушать лекцию достаточно на нее просто прийти. Можно посещать не все, а только некоторые из представленных курсов.

ПЛАНЫ НА ОСЕННИЙ СЕМЕСТР 2008

В осеннем семестре планируется много новых интересных курсов:

- *Юрий Владимирович Матиясевич* расскажет о десятой проблеме Гильберта. (Юрию Владимировичу принадлежит доказательство алгоритмической неразрешимости десятой проблемы. Недавно Юрий Владимирович был избран в действительные члены Российской академии наук.)
- *Эдуард Алексеевич Гирш* выступит с курсом по теории сложности.
- *Александр Шень* прочтет курс по теории информации.
- *Сергей Игоревич Николенко* прочтет курс по теории аукционов.
- Будет также прочтён курс по биоинформатике.

СЕРИЯ КУРСОВ «ЭФФЕКТИВНЫЕ АЛГОРИТМЫ»

Курс «Эффективные алгоритмы» – это всего лишь один из курсов, читающихся в клубе. Рассказать обо всех курсах невозможно, и мы остановились только на одном из них. В следующих семестрах этот курс будет повторен, и появится возможность его послушать.

Курсы «Эффективные алгоритмы» и «Эффективные алгоритмы II» посвящены построению и анализу алгоритмов и рассчитаны на слушателей, знакомых с базовыми результатами в этой области (такими, к примеру, как сортировка, обход графа). Были представлены несколько результатов из классического учебника «Введение в алгоритмы» Кормена, Лейзерсона и Ривеста и из достаточно нового учебника «Алгоритмы» Дасгупты, Пападимитриоу и Вазирани. Большинство же лекций выходило за рамки этих книг. По мнению автора курса, книга «Алгоритмы» достаточно хорошо написана; её черновик все еще можно скачать со следующей страницы: http://www.cs.berkeley.edu/~vazirani/algorithms.html. На amazon.com два упомянутых учебника продаются вместе.

Для таких задач, как проверка результата перемножения матриц, минимальный разрез, минимальное покрывающее дерево, кратчайшие пути до всех вершин графа, проверка равенства нулю многочлена (в ситуации, когда многочлен задан не списком своих коэффициентов), были приведены вероятностные алгоритмы, работающие (асимптотически) быстрее известных детерминированных алгоритмов. К примеру, был приведен вероятностный алгоритм, строящий минимальное покрывающее дерево графа, математическое ожидание времени работы которого линейно. Вопрос о том, существует ли для данной задачи линейный детерминированный алгоритм, до сих пор остается открытым.

Были также приведены несколько известных методов дерандомизации (то есть методов устранения использования случайных чисел из алгоритмов). Приведем простой пример, хорошо иллюстрирующий два таких метода. Рассмотрим граф с m ребрами. Нетрудно показать, что у такого графа всегда найдется двудольный подграф с $m/2$ ребрами. Действительно, рассмотрим случайное подмножество его вершин U, включая в это подмножество каждую вершину независимо с вероятностью 1/2. Легко видеть, что математическое ожидание количества ребер, соединяющих вершины множества U с оставшимися вершинами, равно $m/2$. Значит, существует хотя бы одно такое подмножество вершин U, для которого число таких «пересекающих» ребер не менее $m/2$. По этому простому утверждению легко строится вероятностный алгоритм, находящий двудольный подграф с математическим ожиданием числа ребер хотя бы $m/2$. Приведенный алгоритм можно превратить в детерминированный методом условных вероятностей: на каждом шаге алгоритм будет последовательно добавлять или не добавлять вершину в множество U, так чтобы максимизировать условное математическое ожидание количества «пересекающих» ребер. Несложно показывается, что в итоге будет построен двудольный подграф с нужным количеством ребер.

Рассмотренный вероятностный алгоритм может быть также дерандомизирован методом малого пространства событий. Заметим, что в приведенном нами алгоритме вершины попадали в множество U независимо друг от друга. Независимость необходима для того, чтобы вероятность каждого ребра стать «пересекающим» равнялась ровно 1/2. Для этого, однако, достаточно так называемой независимости по два (когда вероятность того, что любые две случайные переменные приняли какие-то конкретные значения, равна произведению соответствующих двух вероятностей). Это замечание позволяет нам значительно уменьшить размер соответствующего пространства событий, после чего мы можем довольно быстро просто перебрать все возможные значения случайных переменных.

Много внимания было уделено решению NP-трудных задач. Несмотря на то, что эффективных алгоритмов для таких задач не существует, они часто возникают на практике в приложениях. Были рассмотрены алгоритмы, находящие приближенные решения за полиномиальное время для задач о вершинном покрытии, покрытии множествами, кратчайшей общей надпоследовательности, рюкзаке, потоке в сети с несколькими веществами, максимальном разрезе и другими. К примеру, для задач о вершинном покрытии и покрытии множествами работает жадная эвристика: можно показать, что если на каждом шаге покрывать максимальное количество еще не покрытых элементов, то полученное решение будет не очень сильно отличаться по размеру от оптимального. Известно также множество приближенных алгоритмов, основанных на линейном программировании. Например, та же задача о вершинном покрытии может быть легко сформулирована как задача целочисленного линейного программирования, в которой для каждой вершины v исходного графа заводится переменная x_v (напомним, что в то время как задача линейного программирования может быть решена за линейное время, условие целочисленности используемых переменных делает ее NP-трудной). Если v лежит в вершинном покрытии, то $x_v = 1$, в противном же случае $x_v = 0$. Условие на то, что каждое ребро (u,v) графа покрыто, может быть записано следующим образом: $x_u + x_v \geq 1$. Теперь условие $x_v = 0$ или $x_v = 1$ можно заменить на $0 \leq x_v \leq 1$ (такой переход от целочисленных переменных к вещественным обычно называется релаксацией), решить за полиномиальное время полученную задачу линейного программирования, после чего округлить найденные значения переменных x_v. Несложно показать, что найденный таким образом набор значений задаёт вершинное покрытие, вес которого будет не более чем в два раза хуже оптимального.

Было также показано, как можно доказывать нетривиальные верхние оценки для NP-трудных задач. В частности, мы привели сравнительно недавно построенный алгоритм, решающий задачу о максимальном разрезе быстрее полного перебора. Именно, нетрудно видеть, что задача о максимальном разрезе может быть решена за время $O(m2^n)$, где n и m суть количества вершин и ребер графа, соответственно. До недавнего времени не было известно, существует ли алгоритм, работающий (асимптотически) быстрее. Элегантным решением этой задачи является следующее. По исходному графу можно построить вспомогательный граф на трех долях, в каждой из которых будет $2^{n/3}$ вершин, со следующим полезным свойством: в исходном графе существует разрез веса k тогда и только тогда, когда во вспомогательном графе есть 3-клика (то есть полный граф на трех вершинах) веса k. Далее необходимо заметить, что наличие 3-клики в графе без весов может быть проверено при помощи алгоритма умножения матриц: действительно, достаточно возвести матрицу смежности графа в куб и проверить, есть ли ненулевой элемент на диагонали полученной матрицы. Напомним, что если A – матрица смежности графа, то $A^k[i, j]$ равно количеству путей длины k из вершины i в вершину j. Поскольку известны алгоритмы, перемножающие матрицы размера $n \times n$ быстрее чем n^3, получаем выигрыш во времени и для нашего алгоритма. Отметим, однако, что приведенный алгоритм не имеет практической ценности по следующим двум причинам: во-первых, у него экспоненциально не только время работы, но и используемая им память (для хранения матрицы смежности вспомогательного графа), во-вторых, быстрые алгоритмы умножения матриц дают реальный выигрыш только на матрицах огромного размера, которые не помещаются в реальную память.

Также было приведено несколько алгоритмов, обрабатывающих вход по мере поступления. Такие алгоритмы требуются для задач, в которых все входные данные заранее не известны. Такова, к примеру, задача кэширования. Для нее был приведен алгоритм, который подгружает блоки в память не намного чаще, чем некоторый воображаемый алгоритм, который знает всю последовательность запросов наперед.

Рис. 2. Fernando Pedone, курс Distributed Algorithms (Распределенные алгоритмы), обсуждение

Посов Илья Александрович,
аспирант математико-механического факультета СПбГУ.