

Алгоритмическая направленность курса информатики.

Хотя в конце заголовка стоит точка, в настоящее время уместней поставить знак вопроса-- очень большой вопросительный знак. Ведь теперешний курс информатики освещает эту важнейшую её компоненту столь слабо и в таком незначительном объёме, что об алгоритмической культуре учащихся (способности конструировать, обосновывать и оценивать алгоритмы) и, кстати не формируемой и на уроках математики, говорить не приходится (разумеется в позитивном смысле)--нет оснований.

Алгоритмический язык, несмотря на его важную, дисциплинирующую мышление, роль--только язык-средство выражения, представления информации, то есть форма, предназначенная для представления определённого содержания.

Алгоритмический язык, сам по себе, сразу даёт возможность записать простейшие алгоритмы для задач с вырожденным процессом алгоритмизации или готовые алгоритмы, то есть как-то уже описанные, сформулированные с помощью другого языка (в том числе и математического) но чаще бытового или прикладного.

При ориентации на пользователя-потребителя готовых систем это избыточно, а в отношении будущего программиста (творческого работника в среде информатики) -- недостаточно. Каждый умеющий читать -- читатель (потенциальный), однако не каждый, умеющий писать, -- писатель.

Но ведь учат же в школе излагать точно и чётко свои мысли, анализировать произведения, делать выводы, писать сочинения пользуясь данными языковыми средствами. Именно эти мероприятия призваны обеспечить необходимую языковую культуру, хотя направлены для достижения не только этой цели.

Аналогично, для развития алгоритмической культуры учащихся, необходимо включить в программу по информатике обучение процессу алгоритмизации. Но, учитывая, что достаточную общематематическую базу для этого курс математики не обеспечивает, можно поддержать эту часть программы по информатике либо факультативом, либо (что гораздо лучше) внесением соответствующих разделов и методических принципов в курс школьной математики (что обогатит их обоим). Причём сделать это нужно органично вплетая алгоритмическую составляющую на всех этапах обучения математики. В младших классах акцентируя внимание в первую очередь на построение ситуационных моделей при решении текстовых задач, их анализе и, разумеется, последующем их решении. (Тут имеется в виду следующее: текст любой школьной текстовой задачи лишённый количественного итогового вопроса--ситуацию, описанную житейским или предметным языком, иногда сознательно зашумленную (присутствует избыточная информация) либо недостаточно информационно обеспеченную--записать с помощью имеющихся языковых средств и установить ограничения и цели.

То есть заниматься построением математических моделей, начиная разумеется, с самых простых. Затем, подвергая модель анализу указать на возможные варианты её разрешения. (варианты возможных вопросов к модели ситуации). Причём чем выше уровень математического образования, культура математического мышления, тем больший ассортимент ситуаций можно подвергнуть математическому моделированию. Кстати, именно на этом этапе могут продемонстрированы межпредметные связи и философская, познавательская и, наконец, преобразующая сущность этого удивительного инструмента человеческого мышления -- модельного восприятия мира. Но человеческому мышлению и его деятельности присуще и другое: алгоритмичность, технологичность, способность выстраивать все свои действия-- предполагаемые и реальные в виде цепочки последовательных допустимых операций, преобразующих исходный материал (будь-то информация или промышленное сырьё) в конечный продукт с заданными свойствами, удовлетворяющий заданным требованиям.

Высокая разрешающая способность этого свойства возможна лишь при его осознанном, целенаправленном использовании (как это было с законом сохранения энергии в физике, принципом обратной связи в кибернетике, аксиоматическим методом в математике).

И, хотя вроде бы никого не нужно в этом уговаривать, и в высказываниях об информатике авторитетных представителей науки постоянно говорится о важнейшей, основополагающей роли единства этих двух компонент -- модель и алгоритм, на деле почему-то предпочтение отдаётся третьей компоненте информатики -- собственно программированию как ближайшей к результирующему продукту.

Но программирование, заключительный, сложный и самостоятельный этап напрямую закорачивающий

технику преобразующую информацию и конечный продукт всё-таки зачастую лишь надводная часть айсберга информатики и сводить её к программированию, и тем более к изучению языков программирования, пусть даже алгоритмических, неправомерно поскольку эффективность этого завершающего этапа, предсказуемость и надёжность программы напрямую зависит от положенного в её основу алгоритма. И то что курсы информатики используют алгоязыки ничего в этом плане не меняет. Ведь всякий алгоритмический язык при наличии транслятора превращается в язык программирования. Поэтому различие между алгоязыком и языком программирования условно и следовательно курс информатики сведён, правда завуалированно к достаточно хорошему для школы курсу программирования. Но знание алгоритмического языка хоть и даёт возможность писать программы, само по себе ещё не обеспечивает способности конструировать алгоритмы и его недостаточно для формирования алгоритмической культуры мышления.

Мне кажется, что уход от верного решения задачи формирования алгоритмического мышления вызван ещё и тем, что ни теория алгоритмов, ни любой другой раздел информатики в самом широком понимании содержания этого предмета не даёт и не может дать конкретного, окончательного рецепта--алгоритма как проводить процесс алгоритмизации, как строить алгоритм. Ведь построение алгоритма это в конечном счёте искусство решать задачу, проблему, только не экзистенциально, а конструктивно, то есть конечным числом допустимых операций. Особенно трудноосуществимым и далёким это выглядит для средней школы, где набор технических средств(общематематическая база алгоритмизации) кажется весьма скудной.

Но это не так. Ибо есть опыт уже решённых проблем, есть техника исполнения без которой любое искусство немыслимо и которую можно приобрести в процессе обучения, есть, наконец элементарная алгоритмическая база-- языково понятийная база математики и алгоритмический язык. (все эти "есть" разумеется нужно воспринимать как формальные предпосылки, как возможности). Известны приёмы, целенаправленные комбинации способные дать новый результат и есть например типичные ситуации(наряду с нетипичными) с которыми сталкивается всякий постановщик и алгоритмист.

Мой допреподавательский (алгоритмическо-программисткий опыт) и уже преподавательский опыт убеждает, что достаточное количество реальных прикладных (производственных) задач требующих алгоритмизации, и тем более учебных задач, разрешимы с использованием элементарных средств не выходящих за рамки так называемой элементарной (школьной) математики на уровне сложности олимпиадных задач и известной как правило тем кто успешно в математических олимпиадах участвует. (я имею здесь в виду те разделы элементарной математики знакомство с которыми традиционно входит в обязательную программу подготовки к математическим олимпиадам высокого уровня). При этом целенаправленное изложение некоторых традиционно факультативных разделов элементарной математики с примерами постановки и алгоритмизации превращают тяжёлые алгоориетированные задачи в достаточно несложные упражнения.

Только хочу ещё раз заметить, что алгоязыка недостаточно для обеспечения нужд процесса постановки и алгоритмизации. Необходим по сути весь общематематический язык пополненный в целях удобства и необходимости того или иного алгоязыка и языка той предметной области к которому относится задача, то есть по сути то, что А.П. Ершов назвал лексиконом. (Например знак := "присвоение" , "assign" используется в общематематической литературе для введения новых понятий и обозначений; его использование позволяет в безиндексной форме описывать рекурсивные процедуры:

$S:=0;n:=1$ базовый блок

$n:=n+1 ;S:=F(S.n)$ индуктивный блок.

В чём же должна проявляться алгоритмическая направленность школьного курса информатики? На мой взгляд хотя бы в том , что в программе по информатике в явной форме должны быть отражены следующие моменты:

1. Примеры построения моделей ситуационного моделирования в случаях четких и нечётких требований к цели.
2. Примеры, подчёркивающие необходимость алгоритмизации, ярко иллюстрирующие процесс алгоритмизации, специфику вопросов возникающих при этом: требование эффективности алгоритма, его оптимальности по ограничениям: объёма памяти (компактность), быстродействию (учёт факторов влияющих на скорость исполнения операций), оптимизация тела цикла по времени его исполнения и алгоритмические средства её достижения.
3. Методология алгоритмизации. Во всех случаях алгоритмизация идёт от анализа модели с учётом

характеристик исполнительских средств (язык программирования, его возможности, быстродействие выполнения элементарных операций, объём оперативной быстрой и оперативной медленной памяти), возможностей аппарата алгоритмизации (математические средства), границ применимости алгоритма возможных и желаемых. И во всех случаях алгоритмизация заканчивается синтезом готовых алгоритмических самостоятельных или вспомогательных блоков.

Я бы отнёс к вопросам алгоритмизации учёт объёмов входной, выходной и промежуточной информации, подготовку специальных проверочных (тестовых) задач, выявление контрольных точек будущей программы.

4. Необходимо указать простейшие примеры пороговых проблем алгоритмизации, которые требуют либо эвристических подходов, либо частично решаемых (малые размерности), алгоритмически неразрешимых проблем, указать на роль учёта специфики задачи, привести примеры, когда использование специфики позволяет добиться результатов, недостижимых общими методами. Необходимо также привести примеры того как результативность алгоритма и его эффективность зависят от способа представления исходной информации (структуры данных).

Я понимаю избыточность такого подхода в школе, но на любом доступном уровне на уроке или факультативе эти положения желательно сформулировать и продемонстрировать подробно или вскользь, на простых или более сложных примерах (зависит от времени и подготовленности аудитории).

Но самое главное, что надлежит сделать, предваряя в определённой степени все эти пункты, это обеспечить элементную (техническую) базу школьной алгоритмизации.

Я имею в виду понятия и техники из математики, которые наиболее часто используются в процессе решения задач моделирования и алгоритмизации, поскольку опыт у математики в вопросах алгоритмизации огромный и желательно в допустимых в школе пределах с этим опытом знакомить учащихся.

Сюда входят следующие разделы:

- 1.** Язык конечных множеств, отношений и отображений.
 - 2.** Алгебра высказываний и предикатов. Области истинности предикатов. Уравнения и неравенства.
 - 3.** Метод математической индукции и рекурсивные определения.
 - 4.** Повторение на уровне 9-го класса вопросов арифметики целых чисел (Теория делимости, решение сравнений и простейших диофантовых уравнений в целых и неотрицательных целых числах). Свойства целых и дробных частей.
 - 5.** Элементы комбинаторики. Основные правила и приёмы подсчёта.
 - 6.** Последовательности и рекуррентные (рекурсивные) способы их задания. Решение рекуррентных соотношений первого и второго порядков с постоянными коэффициентами (однородных и неоднородных). Решение дробно-линейных рекуррентных соотношений.
 - 7.** Свойства операций \max, \min . Алгебра этих операций. Использование \max, \min в системах неравенств с одной переменной (правило границ).
 - 8.** Системы линейных неравенств с многими переменными. Диогонализация системы неравенств с использованием:
 - а)** Принципа эквивалентного дополнения и исключения;
 - б)** Условия непротиворечивости и последовательного исключения переменных.
- Задание многомерных областей при помощи системы неравенств.
Преобразования многомерных областей.
- 9.** Операторы разности и суммирования. Свойства и приёмы суммирования. Подсчёт количества целых точек в областях заданных системами неравенств.
 - 10.** Элементарные методы оптимизации.
 - а)** Метод последовательной минимизации (максимизации). Метод наименьших квадратов.
 - б)** Базовые неравенства и их использования для решения задач оптимизации.
 - с)** Решение задач линейного программирования методом последовательной оптимизации
 - 11.** Решение линейных целочисленных неравенств.

Аркадий Маркович Альт

08.12. 1987

Обнаружено в архивах академика А.П. Ершова- документ 255-80