Паньгина Нина Николаевна

ПОДГОТОВКА УЧЕНИКОВ К ОЛИМПИАДАМ ПО ИНФОРМАТИКЕ

Олимпиады по информатике являются по сути своей олимпиадами по программированию. Решение олимпиадных задач представляет собой вполне самостоятельный учебный раздел с обширными теоретической и практической частями.

Шесть лет на базе школы-лицея № 8 г. Сосновый Бор ведется подготовка школьников к олимпиадам различного уровня. В течение пяти лет учащиеся лицея являются победителями областных олимпиад по программированию, участниками и призерами Всероссийских олимпиад.

Решения олимпиадных задач разных уровней, начиная с районного и заканчивая международным, базируются на вполне определенных алгоритмах, широко известных в математике и информатике, и, чтобы успешно решать их, необходимо прежде всего освоить эти алгоритмы, увидеть их и умело применить в предлагаемых заданиях. А уж если не знаешь, то нужно суметь их придумать, изобрести. Но знакомство с этими алгоритмами чаще всего происходит только в вузе, и это вполне объяснимо, так как их освоение требует знаний некоторых разделов высшей математики, не входящих в программу средней школы.

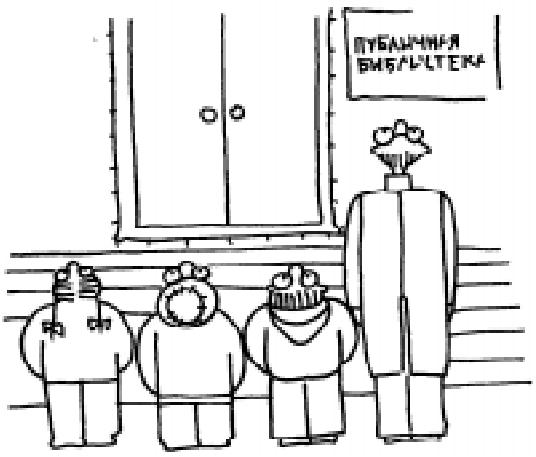
Вопрос 1. *Так можно ли подготовить школьников к олимпиадам?*

Можно, но только не в рамках базовой программы. Притом, почти необходимое условие, чтобы эти школьники занимались дополнительно математикой (лучше, если это классы с углубленным изучением математики). Неслучайно наши российские выдающиеся школьники в области информатики занимались столь же успешно и математикой:

– Виктор Баргачев, 2-кратный абсолютный чемпион мира по информатике среди школьников, является призером Международной олимпиады по математике (серебряная медаль);

– Николай Дуров, 4-кратный призер Международных олимпиад по информатике (три серебряные и одна золотая медаль), является обладателем двух золотых медалей Международных олимпиад по математике;

– Владимир Мартьянов из Нижнего Новгорода (2-кратный абсолютный чемпион мира по информатике) являлся победите-



Так можно ли подготовить школьников к олимпиадам?

лем зональных Российских олимпиад по математике и физике.

По Ленинградской области можно привести для примера Стратонникова Алексея, Потапова Алексея, Ананьева Артема, Паньгина Андрея.

С какого же возраста необходимо начинать готовить школьников?

Вопрос 2. *С какого же возраста необходимо начинать готовить школьников?*

Чем раньше, тем лучше, но в разумных пределах. Для областных олимпиад достаточно начинать готовить с 7–8 класса, для российских – желательно начинать с 5–6 класса.

Например, чемпион Москвы по программированию среди школьников 1998 года Петр Митричев – ученик 7 класса, он же явился самым молодым участником IX Всероссийской олимпиады школьников по информатике и участником тренировочных сборов по подготовке команды России к Международной олимпиаде.

Вопрос 3. *В какой форме проводить занятия по подготовке к олимпиадам?*

В виде факультативов или спецкурсов. Для хорошей подготовки ученика важно, в первую очередь, «не только наполнить чашу знаний, но и зажечь факел». Иными словами – привить интерес!

По опыту нашего лицея этому способствует следующее.

1. Межпредметные связи. В лицее уже пять лет ведется спецкурс «Математика на компьютерах» для 5–6 классов. Учащиеся знакомятся в наглядной форме с математическими понятиями (например, анимационной разверткой куба, координатной плоскостью с целочисленной сеткой), учатся творчески подходить к решению стандартных задач (например, с помощью графического представления на компьютере решаются задачи переливания и взвешивания).

Развитию интереса способствуют уроки по теме «Моделирование движения» от простейшего движения бильярдного шара до сложной картины линий напряженности электрического поля двух зарядов. Особенно производит впечатление моделирование на компьютере «полета бумажного самолетика» и демонстрация этого полета наяву. Также интересна задача моделирования движения спутника и определение первой, второй космической скорости, времени полета ракеты «Восток» с Юрием Гагариным вокруг Земли (это событие должен знать каждый!).

Такие уроки проводятся дифференцированно, в зависимости от степени подготовленности учащихся.

2. Для 7–8 классов организуется в течение первых двух-трех недель летних каникул школьный лагерь «Интеллект», где ребята занимаются информатикой по специальной программе.

Например, программа «Лабиринт» включает темы:

- построение лабиринта (интересный алгоритм генерации случайного лабиринта различной сложности);
- передвижение по лабиринту с помощью управляющих клавиш;
- поиск выхода из лабиринта, начиная с простейшего, по правилу левой руки (придерживаясь одной стороны лабиринта);
- поиск кратчайшего пути в лабиринте.

Эта программа дает толчок учащимся для разработки собственных игр – «ходилок» и «стрелялок». Главное – осознание того, что они сами могут это создавать!

3. Для 10 классов организуется летняя производственная практика на различных предприятиях г. Сосновый Бор в течение одного месяца. Учащиеся приобретают навыки решения практических задач, закрепляют знания по работе с базами данных, бухгалтерскими офисными приложениями, самостоятельно создают программы по заданиям кураторов, при-

обретают дополнительные знания (например, при работе с компьютерными сетями, операционной системой Unix). Похвальный отзыв с предприятия – обычный результат такой работы.

4. Собственные программы ребята демонстрируют на традиционной школьной конференции по информатике «Мы и компьютер» (в этом году была уже шестая конференция). Лучшие из программных разработок ребята ежегодно представляют на Международной конференции «Школьная информатика и проблемы устойчивого развития», проходящей в Санкт-Петербурге, где работы ребят оцениваются дипломами I и II степени.

Отдельные программы используются как учебный материал.

5. Для группы «компьютерных фанатов» (в хорошем смысле, а не в смысле, например, компьютерных игр) ведутся специально направленные факультативы и спецкурсы по темам: изучение дополнительного языка программирования, алгоритмизация нестандартных задач, олимпиадные задачи, теория графов и т.п.

Вопрос 4. *Сколько человек должно быть на занятиях специально ориентированных факультативов?*

Группы по 6 – 12 человек. При работе на конкретный результат (то есть при подготовке к городской, областной или российской олимпиаде), должна быть группа от 3 до 6 человек. Эти занятия представляют собой тренировки, и тут преподаватель больше выступает в роли тренера, а не учителя. Как проходит занятие?

- Называется тема.
- Перечисляются задачи на данную тему.
- Выбирается одна из наиболее популярных или интересных задач.
- Устно совместно с ребятами обсуждается алгоритм решения.
- Ребята пишут программу, преподаватель фиксирует время, оценивает реализацию решений, помогает искать ошибки, указывает на недочеты по эффективности (количество операций, использование оперативной памяти, время решения).

Вопрос 5. *Какие темы необходимо изучать на занятиях по подготовке к олимпиадам?*

Всего не предусмотреть, но можно выделить следующие группы алгоритмов:

1. Алгоритмы над целыми числами.
2. Рекурсия.
3. Сортировка.
4. Переборные задачи.
5. Геометрические задачи.
6. Численные методы.
7. Статистическое моделирование.
8. Динамическое программирование.
9. Графы и деревья.
10. Текстовые преобразования.

Рассмотрим более подробно темы, необходимые для изучения при подготовке к олимпиадам. Теоретические занятия должны включать в себя определения, утверждения (в некоторых случаях обязательно с доказательствами).

1. Алгоритмы над целыми числами

1.1. Делимость

При изучении этой темы необходимо дать определения делителя, кратного, простых и взаимно простых чисел, привести утверждения с доказательствами о делимости суммы и разности двух чисел, о делимости произведения.

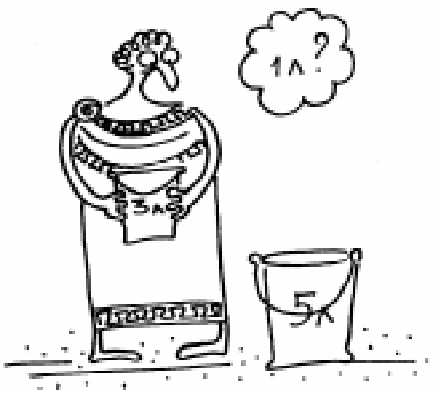
1.2. Первая модификация алгоритма Евклида (с вычитанием)

$$\text{НОД}(a,b) = \begin{cases} \text{НОД}(a, b-a) & при \quad b > a \\ \text{НОД}(a-b, b) & при \quad a > b \\ a & при \quad b = a \end{cases}$$

Последовательно уменьшая числа (большее заменяя разностью чисел) до тех пор, пока одно из них станет нулем, придем к наибольшему общему делителю этих чисел.

Задачи, которые необходимо разобрать при изучении данной темы:

- Определить, являются ли несколько чисел взаимно простыми.
- Сократить дробь (числитель и знаменатель дроби вводятся).
- Написать программу «Калькулятор» в простых дробях.



...необходимо разобрать ... задачи на переливание...

1.3. Вторая модификация алгоритма Евклида, использующая деление с остатком

$$\text{НОД}(a,b) = \begin{cases} \text{НОД}(a,r) & при \quad b > a \\ \text{НОД}(r,b) & при \quad a > b \end{cases}$$

где r – остаток от деления большего из чисел a и b на меньшее.

Таким образом, уменьшая числа, получим наибольший общий делитель как последний не равный нулю остаток.

Задача. Прямоугольник с длинами сторон a и b, где a и b – натуральные числа, разрезать на квадраты максимальной площади. Определить размер квадратов и их общее количество.

1.4. Диофантовы уравнения

Используя утверждение о представлении наибольшего общего делителя двух чисел в виде линейной комбинации этих чисел, приходим к утверждению о разрешимости в целых числах уравнения вида $ax + by = c$ (диофантово уравнение).

Задачи, которые необходимо разобрать при изучении данной темы:

- Задачи на размен денег монетами определенного достоинства.
- Задачи на переливание.

При изучении алгоритма решения задачи на переливание представляется полезным использование вспомогательной демонстрационно-обучающей программы «Переливайка», с помощью которой учащиеся могут на практике применить изученный алгоритм. Необходимым является закрепление этого алгоритма для нахождения хотя бы одного решения диофантова уравнения. Затем дается общий алгоритм решения диофантова уравнения с формулами для нахождения всего множества решений.

1.5. Простые числа

С простыми числами связано множество олимпиадных задач разных уровней. Классический метод для нахождения простых чисел – решето Эратосфена. С этим алгоритмом связаны решения следующих задач:

- Числа-близнецы.
- Совершенные числа.
- Скатерть Улама.
- Дружественные числа и т.п.

1.6. «Длинная» арифметика

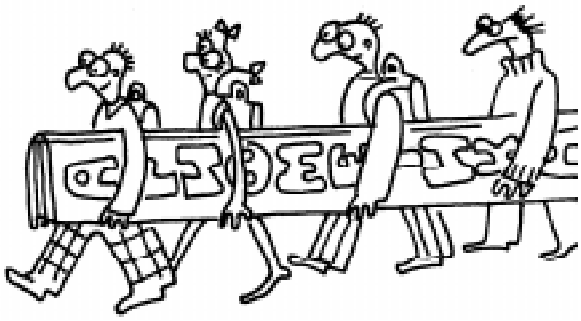
Задачи на «длинную» арифметику возникают тогда, когда в стандартных типах данных (целые или длинные целые) не представляется возможным хранить результат – настолько он велик. В этом случае используется прием хранения длинных чисел в виде массива цифр, а чтобы выполнять арифметические действия с такими числами, необходимо написать специальные процедуры сложения, умножения, деления длинных чисел.

Типовые задачи:

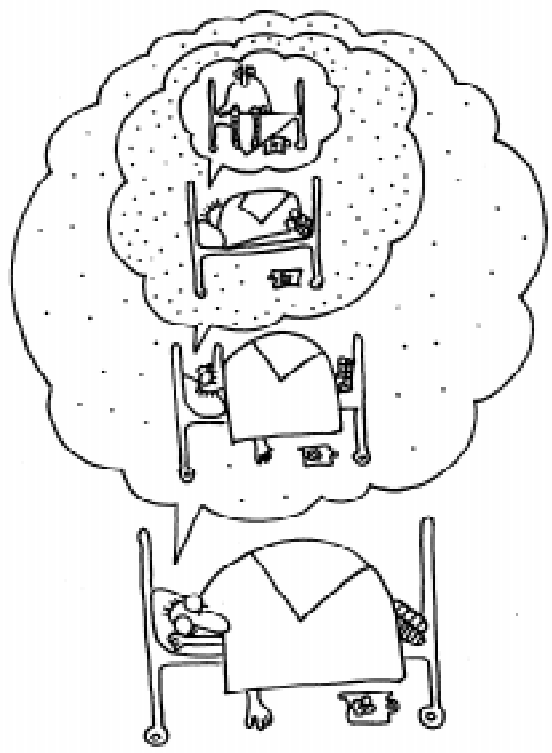
- Найти число N! (N-факториал).
- Определить период дроби.

2. Рекурсия

Почти в любой книге по программированию затрагивают рекурсию, рекурсивные процедуры и функции. Но очень мало книг, где тема рекурсии разбиралась бы подробно. Традиционно начинается рассмотрение рекурсивной функции с на-



Задачи на «длинную» арифметику возникают...



Следующий этап – рекурсивные рисунки...

хождения факториала числа. Короткая функция в одну строчку, но непонятно, зачем считать факториал рекурсивно, если он и так считается элементарно просто в одном цикле с параметром.

Чем младше ученики, тем важнее для них принципы наглядности и простоты. Введение в рекурсию можно начинать с известной считалки про «10 негритят».

Следующий этап – рекурсивные рисунки. Начинаем с простейших рисунков, например, рисования упрощенной «матрешки».

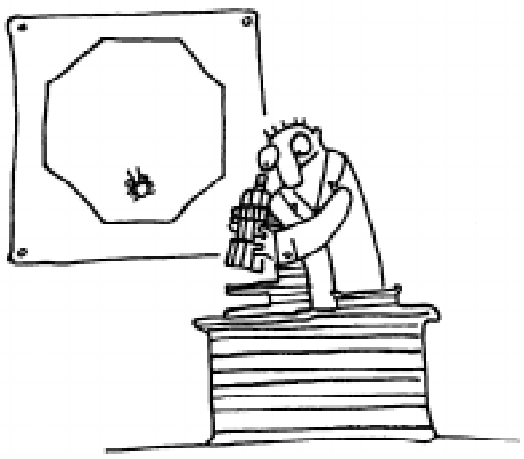
Далее переходим к более красивым и более сложным рисункам «Снежинка», «Парад планет», «Веточка». И вот тут проявляется творчество ребят:
– снежинки, падающие по всему экрану;
– веточки разного цвета, разной пушистости и разного размера (в длину ветки вносится случайная составляющая).

Большой интерес представляют для ребят фрактальные множества: салфетка и скатерть Серпинского, модель Мандельброта человеческого легкого, фрактал Хартера-Хейтуэя (известен нам как ломаная Дракона), кривые Гильберта, снежинка Коха. Рекурсия здесь настолько естественна, что ни у кого не возникает вопроса, можно ли без нее обойтись. И лишь теперь, когда прелесть и красота рекурсии не вызывает ни у кого сомнений, можно переходить к задачам, которые по своему определению являются рекурсивными. Это следующие задачи: факториал числа, *N*-я степень числа, НОД(*a*,*b*), функция Аккермана, числа Фибоначчи, перевод чисел из десятичной системы счисления в двоичную, нахождение максимума и минимума в массиве. Многие задачи, оказывается, можно делать с помощью рекурсии. Главное, что появляется видение рекурсии в задачах, ранее решенных не рекурсивно. Например, в задаче о разрезании прямоугольника на квадраты максимальной площади предлагается сделать наглядное графическое сопровождение с помощью рекурсии. И уж совсем «плохо» без рекурсии при решении таких задач, как «Ханойская башня». В данном случае осознать это и на практике освоить рекурсивный алгоритм решения данной задачи помогает демонстрационно-обучающая программа «Ханойская башня».

3. Сортировка

Без знаний алгоритмов сортировки никак нельзя спокойно чувствовать себя

Без знаний алгоритмов сортировки никак нельзя спокойно чувствовать себя...



...содержится ли точка внутри произвольного многоугольника.м

на олимпиадах различного уровня. Задачи могут формулироваться как явно с упоминанием алгоритма сортировки, так и подразумевая внутри решения использование алгоритма сортировки. Два простейших алгоритма, которые необходимо знать, – это «пузырек» и сортировка выбором (с помощью поиска последовательных минимумов). При небольших объемах данных вполне хватает этих двух методов, но при работе с данными, измеряемыми десятками и сотнями Кбайт, необходимо знание какой-нибудь из быстрых сортировок. Предпочтение можно отдать быстрой сортировке Хоара, которая реализуется рекурсивно. Необходимо также знать некоторые приемы при упорядочивании данных в больших файлах. Весь файл разбивается на некоторое количество кусков, которые можно отсортировать с помощью QSORT, а затем произвести слияние отсортированных файлов уже без программы сортировки (кстати, это очень полезная с технической точки зрения задача, требующая аккуратности и хорошей техники программирования).

4. Перебор вариантов

Задачи перебора составляют огромный класс олимпиадных задач. Начинается перебор с простейшего поиска минимума и максимума в одномерном массиве или поиска элемента с заданными свойствами. Далее идет перебор пар элементов, использующий вложенный цикл, затем перебор троек элементов, использующий тройной цикл (как, например, в задаче о поиске трех точек на плоскости среди заданных N точек, которые образуют треугольник с максимальной площадью). А если перебирать сочетания из 4, 5, 6 и т.д. элементов? Сколько же необходимо циклов? Оказывается, существуют алгоритмы, существенно сокращающие перебор.

Довольно длинный по времени выполнения, но один из самых универсальных алгоритмов перебора – перебор с возвратом или «бектрекинг». Программируется рекурсивно. Лучше всего объясняется на следующих задачах:

- Ходом коня обойти шахматную доску NxM.
- Расставить 8 ферзей на шахматной доске размера 8x8 так, чтобы они не угрожали друг другу.
- Найти выход из лабиринта и т.п.

Необходимо также объяснять учащимся, что в случае слишком долгого времени выполнения программы, необходимо ограничивать «бектрекинг». Например, в задаче про шахматы при размере доски 8x8 выполняется программа уже довольно долго. Можно и даже нужно применять симметрию, зеркальное отображение, выполняя задание для половины или четверти шахматной доски.

Классической задачей на «бектрекинг» является задача о «рюкзаке», затем можно рассматривать производные от этой задачи:

- Разбить N чисел на два подмножества, наиболее близких по сумме.
- Из заданного множества слов выбрать максимальную подпоследовательность слов в словарном порядке.

С перебором связаны и комбинаторные задачи. Следует освоить алгоритмы создания лексикографического упорядочения.

5. Задачи по геометрии

5.1. Обязательный минимум, который необходимо знать при решении задач по геометрии, включает в себя следующее:

• Уметь привести уравнение прямой, проходящей через две точки к виду $Ax + By + C = 0$.
• Уметь определить, принадлежит ли точка прямой или отрезку.
• Уметь определить, пересекаются ли две прямые, и, если пересекаются, то определить их точку пересечения (для этого вычислять определитель 2-го порядка).
• Уметь написать уравнение перпендикулярной прямой или определить, перпендикулярна ли одна прямая другой (использование свойства о равенстве нулю скалярного произведения перпендикулярных векторов).
• Уметь вычислять расстояние от точки до прямой.

5.2. В дополнение к этому для решения задач областного и Всероссийского уровня необходимы знания начального курса аналитической геометрии: скалярное и векторное произведение векторов, выражение их через координаты. Скалярное произведение используется для нахождения угла (или косинуса угла) при построении, например, выпуклой оболочки заданных точек. Векторное произведение двух векторов используется, во-первых, для вычисления площадей многоугольников, а во-вторых, для определения направления в соответствующих задачах. Эти знания необходимы также для того, чтобы решать такую классическую олимпиадную задачу: содержится ли точка внутри произвольного многоугольника.

Необходимо обратить внимание учащихся на многозначность понятий, которые можно использовать в различных задачах. Например, для понятия векторное произведение:
– длина векторного произведения определяет площадь параллелограмма, построенного на заданных векторах;
– нулевое значение – параллельность векторов;
– знак означает, что один вектор расположен «слева» или «справа» относительно другого;
– результирующий вектор определяет нормаль к плоскости, которой параллельны заданные векторы.

6. Численные методы

6.1. Метод дихотомии

Для поиска данных в упорядоченном множестве используется метод дихотомии (или деления пополам) с последующей проверкой искомых свойств. Этот метод, который более привычен для нахождения приближенных значений корней нелинейных уравнений, можно использовать для поиска среди «огромного» количества данных. В этом случае более употребим термин «бинарный поиск». Типовая задача – найти заданное слово в словаре.

В общем случае, данный метод является оптимальным.

6.2. Решение систем линейных уравнений: метод Крамера, метод Гаусса.

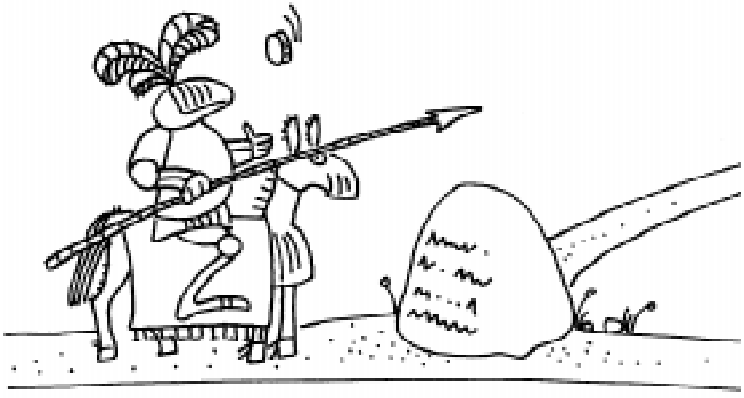
Обычно в задачах число уравнений не более трех. Но знание общего случая, вероятно, пригодится.

Задача (областная олимпиада 1994 г).

Вводится в виде строчки уравнение химической реакции. Необходимо в нем расставить коэффициенты.

7. Статистическое моделирование (метод Монте-Карло)

Очень полезным является знание данного метода:
• Для моделирования игровых вероятностных ситуаций (бросание монеты или кубика, блуждания). Именно «игровая» или



...для моделирования игровых ... ситуаций (... блуждания)...

связанная с чем-то знакомым (известным, «бытовым») формулировка задачи помогает в понимании метода, овладении понятием вероятности. Можно предложить учащимся интересные задачи:
– чего больше: сократимых или несократимых дробей? (Математическая формулировка: оценить вероятность того, что наудачу взятая дробь несократима);
– лучшее пари для простаков («Квант» № 5, 1987);
– комбинаторные задачи.

- Для определения площадей фигур, когда затруднены аналитические решения.

Так, в задаче (областная олимпиада 1999 года) по нахождению площади пересечения трех окружностей метод Монте-Карло можно использовать в качестве альтернативного прямому аналитическому методу, так как очень трудно вывести решение, используя тригонометрические соотношения. Наилучший путь – это «использовать геометрию» для анализа частных случаев (нет пересечения, пересечение в одной точке, одна окружность внутри другой), а метод Монте-Карло – для общего случая.

8. Динамическое программирование

Бывают случаи, когда невозможно решить задачу полным перебором вариантов из-за их огромного количества. Но если на каждом шаге задачу можно разбить на более простые и найти оптимальное решение для них, не рассматривая всех решений общей задачи, а затем найденные решения применить для следующих шагов, то это и означает, что в данном случае применим принцип динамического программирования.

Данный принцип легче всего воспринимается на конкретных задачах. Но объяснение его можно начать с занимательной истории про золотую лестницу Фараона (изложенную в журнале «Квант» № 10, 1991), а затем перейти к следующим задачам:

- Классическая задача о нахождении такого кратчайшего пути в числовой матрице A размерности NxN из $A(1,1)$ в $A(n,n)$, в котором сумма цифр в клетках была минимальной или максимальной.
- Нахождение общей подстроки максимальной длины (задача, возникшая из современной реальной задачи молекулярной биологии об общей генетической информации в молекулах ДНК).

9. Графы и деревья

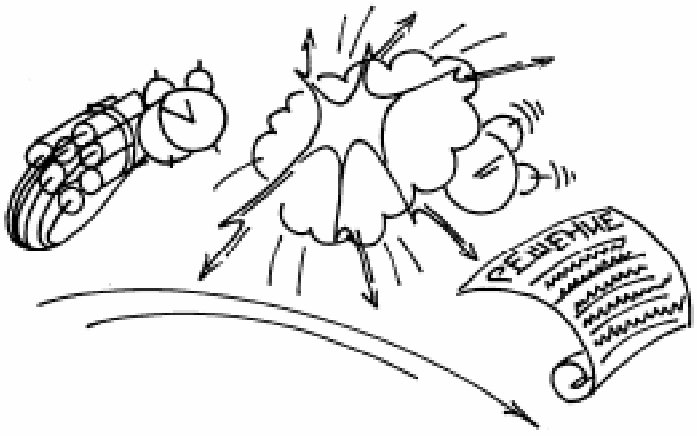
Очень важная тема. Практически ни одна олимпиада Российского уровня не обходится без использования какого-либо алгоритма на графах. Необходимо ввести понятие графа через вершины и ребра, понятие ориентированных или неориентированных графов, разобрать методы представления графов в виде матрицы смежности, матрицы инциденций, списка связей.

Разобрать основные алгоритмы на графах:
– поиск в глубину (иными словами «бектрекинг» или перебор с возвратом);
– поиск в ширину (или метод заливки);
– алгоритм Дейкстры для поиска кратчайших путей в графе из заданной вершины во все остальные;
– алгоритм Флойда для поиска кратчайших путей в графе между всеми парами вершин.

10. Текстовые преобразования

Задачи, в которых необходимо проводить разбор строки, порой дополнительно предполагают работу с файлами, графическими построениями (подобие команд текстового или графического редактора) или использование каких-либо приемов программирования (например, представление «стека» в виде символьной строки в задаче о правильном скобочном выражении для нескольких типов скобок).

Представлен широкий, но далеко не полный набор тем, охватываемых олимпиадными задачами. Во многом это зависит от пристрастий организаторов олимпиад различных уровней (теоретические туры, задачи с использованием игровых стратегий, задачи прикладного характера).

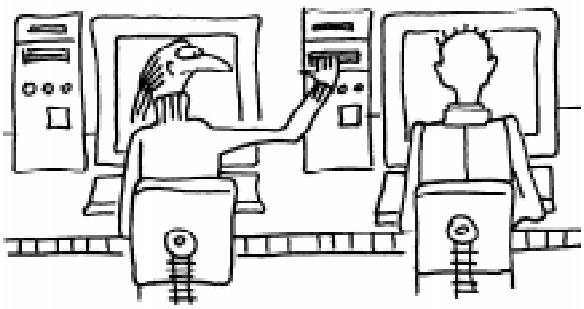


Если для задачи указано ограничение по времени решения...

Вопрос 6. *Что можно посоветовать при решении задач на олимпиаде?*

Предполагая, что теория алгоритмов «проштудирована» и основы программирования освоены, участнику соревнования можно посоветовать следующее:

1. Оцените и распределите свои силы, знания и время в соответствии со сложностью предлагаемых задач.
2. Обеспечьте «про запас» определенный «гарантированный» (по вашему мнению) набор баллов на простых задачах. Это не означает, что сложные задачи надо оставлять на последний момент. При обдумывании их решения может быть найден «хороший» алгоритм, и успехом послужит соответственно большая доля баллов.
3. Если задача трудна для вас или образуется дефицит времени для решения, запрограммируйте простые частные случаи (вдруг некоторые тесты пройдут), либо используйте какой-нибудь неоптимальный, например, переборный алгоритм.
4. Если для задачи указано ограничение по времени решения, а ваш переборный алгоритм не укладывается в это время, то вставьте в алгоритм таймер, который при приближении к критическому времени заканчивает задачу и выводит наилучший найденный переборный вариант (возможно, он будет правильным).
5. Немаловажен выбор алгоритмического языка, например, в Basic-программе легче работать с графикой, а у Pascal-программы – преимущество быстродействия.
6. Строго соблюдайте указанные в условии задачи форматы ввода-вывода: «автотестировщик» может неправильно воспринять лишний пробел, или исходные тесты содержат запятую там, где вы предусматривали во вводе своей задачи пробел (особенности операторов ввода Pascal и Basic).
7. В основном, любая программа имеет структуру:
   – ввод данных;
   – расчетный блок;
   – вывод результата.
8. Оформите и убедитесь, что ввод-вывод правильный (с возможным анализом ошибочных данных) прежде, чем программировать расчетный блок.
9. Поставьте ключи компилятора для обработки всевозможных ошибок (полезно при отладке), а перед сдачей задачи отключите эти опции (для игнорирования компилятором возможных ошибок тестирования, которые не сказываются на решении).
10. Проверьте задачу для заданных максимальных ограничений (например, если дано $n \le 1000$, проверьте для $n = 1000$). Следите за типом объявляемых переменных (предпочтительно для описания целочисленных переменных использовать тип «длинное целое»).
11. Разумно используйте апелляцию. Возможна двусмысленность понимания задачи. В жюри тоже люди, и вы можете отстоять свою позицию, убедить их добавить вам желанные баллы.
12. В крайнем случае, если вы «cool



...запустите программу «соседа»...

hacker» (крутой специалист по системной части), запустите программу «соседа» как свою (может, никто не заметит).
13. Разрешается иметь свои шпаргалки (не в электронном виде) или воспользоваться «свободным» временем для изучения справочной литературы.

Некоторые из перечисленных советов являются «вредными», соответствуют принципам «авось, небось и как-нибудь». Просьба воспринимать их критически, но они жизненны.

Вопрос 7. *Какой литературой необходимо пользоваться при подготовке к олимпиадам?*

Данной статьей, а также следующими источниками.

1. А. Шень. Программирование: теоремы и задачи, М.: МЦНМО,1995.
2. А.Л. Брудно, Л.И. Каплан. Московские олимпиады по программированию, М.: Наука, 1990.
3. В.А. Дагене, Г.К. Григас. 100 задач по программированию, М.: Просвещение, 1993.
4. В.М. Бондарев, В.И. Рублинецкий, Е.Г. Качко. Основы программирования, Харьков: Фолио, 1997.
5. В.М. Кирюхин, А.В. Лапунов, С.М.Окулов. Задачи по информатике. Международные олимпиады, М.: ABF, 1996.
6. Н.М. Бадин, С.Г. Волченков, Н.Л.Дашниц. Ярославские олимпиады по информатике, Ярославль, 1995.
7. С.М. Окулов. Конспекты занятий по информатике (алгоритмы на графах), Киров,1996.
8. А.В. Алексеев. Олимпиады школьников по информатике. Красноярское книжное иэдательство,1995.
9. А.С. Сипин, А.И. Дунаев. Областные олимпиады по информатике, Вологда, 1994.
10. В. Пинаев. Городская олимпиада по программированию, Рыбинск, 1998.
11. С.А. Абрамов. Математические построения и программирование, 1987.

НАШИ АВТОРЫ

Паньгина Нина Николаевна,
преподаватель ОИиВТ
школы-лицея № 8, г. Сосновый Бор.