«Алгоритмы целочисленной арифметики»: факультативное занятие

В. П. Лактина,
учитель информатики высшей категории
гимназии № 8 г. Витебска

Тема «Алгоритмизация и программирование» в школьном курсе информатики является наиболее сложной для большинства учащихся. Но для «выращивания» будущих программистов этого материала очень мало. Мотивированным школьникам и заинтересованному учителю на помощь приходит факультатив «Алгоритмизация и программирование» для 5–11 классов (авторы А. Б. Кондратович, В. П. Лактина). Работаю по этой программе уже много лет. Цель – развитие способностей учащихся, их мышления, логики, памяти, познавательных и практических умений; первостепенное значение при этом придаю формированию способности к самообразованию.

Эффективность данного факультативного курса подтверждают достижения моих учеников на олимпиадах высокого уровня: 73 диплома на заключительном этапе республиканских олимпиад по информатике, 39 дипломов на открытых Российских олимпиадах в Санкт-Петербурге и Москве, 7 медалей на Международных олимпиадах по информатике, проходивших в Тайване (2014 г.), Казахстане (2015 г.), Татарстане (2016 г.), Иране (2017 г.). Отмечу, что эффективность дипломами не ограничивается: за время учебы в университете все бывшие олимпиадники быстро становятся высококлассными специалистами в сфере разработки программного обеспечения и в дальнейшем строят блестящую профессиональную карьеру.

В представленной **разработке факультативного занятия по теме «Целочисленная арифметика»** отражены мои подходы к обучению программированию учащихся средних классов. Урок проникнут межпредметными связями, исследовательской направленностью и способствует интенсивному развитию различных видов мышления учащихся. Для активизации познавательной деятельности учащихся использую проблемный метод обучения. Отмечу, что в школьном курсе информатики 8-го класса алгоритмы целочисленной арифметики представлены минимальным набором: НОД и поиск делителей числа. На факультативных занятиях по данной теме материал изучается гораздо шире и глубже, что дает возможность повышения уровня учебно-познавательной компетенции учащихся, дополнительного развития их способностей, расширения возможностей образовательных стандартов.

Факультативное занятие (2-х часовое) по информатике по теме «Алгоритмы целочисленной арифметики»

Цели урока:

- закрепление материала предыдущего урока;
- формирование навыков и умений составления структурных программ для рения практических задач по теме «целочисленная арифметика»;
- развитие познавательного интереса, логического и алгоритмического мышления, навыков самоконтроля, ответственности, внимания;
- освоение различных методов решения задач, реализуемых на языке программирования;
- углубление знаний, умений и навыков решения задач по программированию и алгоритмизации.

Тип урока: урок усвоения новых знаний.

Учащиеся должны знать:

- ✓ алгоритм нахождения делителей натурального числа,
- ✓ алгоритм проверки простое ли число,
- ✓ алгоритм Решета Эратосфена

Учащиеся должны уметь выполнять практические задачи с использованием изученных алгоритмов.

Программное и методическое обеспечение урока: система программирования Pascal, интернет на ученических компьютерах.

Техническое обеспечение урока: компьютеры.

Ход урока

1. ***Проверка и закрепление знаний и умений предыдущего урока***

К доске вызываю два человека написать решение домашних задач:

1) Сайт acmp.ru №383 «Красивые числа-2»:

Будем называть число красивым, если сумма его цифр делится на количество цифр в нем. Необходимо найти N-ое в порядке возрастания красивое число. (1 <= N <= 100 000)

Пример ввода:	**Пример вывода:**
1	1
15	20

2) Сайт dl.gsu.by раздел «Методы алгоритмизации» задача «Взаимно простые тройки»:

Дано N различных чисел. Определить, какое количество троек из этого набора являются попарно взаимно простыми.

Формат ввода:
N – количество чисел. 1<=N<=100
Последовательность из N чисел. 2<=A[i]<=1000

Пример ввода:	**Пример вывода:**
5 2 4 7 14 9	2

С остальными учащимися провожу фронтальный опрос по теме предыдущего занятия:

- ✓ Какие числа называют четными? Нечетными? Как написать в команде ветвления условие проверки на четность?
- ✓ Что называется наибольшим общим делителем двух натуральных чисел? Рассказать функцию нахождения НОД двух чисел.
- ✓ Какие числа называют взаимно простыми?
- ✓ Какое число называют кратным данного числа? Как получить наименьшее общее кратное двух чисел?
- ✓ Как в программе найти сумму цифр натурального числа и количество цифр?

Выясняю, прошли ли у учащихся в домашнем задании все тесты. Обсуждаем правильность выполнения домашнего задания (на доске), выявляем проблемы, с которыми столкнулись учащиеся при выполнении домашнего задания. Даю рекомендации по их устранению.

2. *Актуализация знаний учащихся на изучение учебного материала. Объяснение нового материала. Составление и реализация алгоритмов (метод проблемного изложения в сочетании с частично-поисковым методом, фронтальная форма работы)*

– Что называют делителем числа? Как найти все делители натурального числа?

Задача 1. Найти все делители натурального числа x ($1<x \le 10^9$).

Обычно ребята предлагают нерациональный алгоритм, который записываем и разбираем его недостатки:

```
Write (1, ‘ ‘, x);
For d:=2 to x div 2 do
 if x mod d = 0 then write (‘ ‘, d);
```

– Какова сложность выполнения данного алгоритма? Успеет ли он при $x=10^9$ за 1 секунду найти все делители? (Ответ: O(x/2), следовательно, не успеет).

– Как усовершенствовать алгоритм?

– Заметим особенность, что все делители (кроме корня) у целого числа парные. Выпишем все пары делителей, например у числа 100:

1, 100 **2**, 50 **4**, 25 **5**, 20 **10**, 10

Сделаем вывод, что искать делители у числа нужно только до его корня.

```
write (1, ‘ ‘,x );
d:=2;
while  int64(d)*d < x do begin
        if   x mod d = 0  then   write (‘ ‘, d, ‘ ‘, x div d);
        d:=d+1;
end;
if  int64(d)*d=x  then  write (‘ ‘, d);
```

Проверяем $\sqrt{x}$ отдельно, т.к. у него нет пары

– Какова теперь сложность выполнения данного алгоритма? Успеет ли он при X=10^9 за 1 секунду найти все делители? (Ответ: $O(\sqrt{x})$, следовательно, успеет за 1 сек.)

– Какие числа называются простыми? Какие составными?
– Как определить простое ли число?

Задача 2. Составить функцию, определяющую, является ли натуральное число x простым. ($1 \le x \le 10^9$)

Ребята обычно предлагают воспользоваться *задачей 1*, подсчитав количество делителей. Предлагаю найти пути усовершенствования алгоритма. Замечаем, что:

а) простое число не может быть четным (за исключением 2),

б) нечетное число не может иметь четных делителей,

в) если нашли хоть один делитель, то число – составное и можно остановить цикл.

Учитывая замечания, составляем функцию:

```
Function Prost (x: longint): boolean;
Var d: longint;
Begin
        If  x mod 2 =0 then Prost:=(x=2)
        else begin
                d:=3;
                while  (int64(d)*d<=x)  and  (x mod d <> 0)  do
                d:= d+2;
                Prost:= (int64(d)*d > x) and (x<> 1);
              end;
end;
```

Из четных чисел простое только 2

У нечетных чисел делители тоже нечетные, ищем их с 3

Пока делитель d не перескочил $\sqrt{x}$ и x не делится на d

– Как найти все простые числа на заданном целочисленном промежутке?

Задача 3. Найти все простые числа на промежутке от 2 до N ($N \le 10^6$).

Ребята обычно предлагают в цикле воспользоваться функцией Prost из *задачи 2*. Выясняем сложность такого алгоритма: N*$\sqrt{N}$. При N=10^6 получим 1 млрд. действий. Следовательно, надо искать более быстрое решение.

– Слышал ли кто-нибудь из вас о Решете Эратосфена?

Выписываю на доске в ряд все числа от 1 до 27 и показываю принцип Решета Эратосфена:

вычеркиваю 1, вычеркиваю все числа кратные 2, кратные 3, кратные 5 (кроме их самих). Остались не вычеркнутыми только простые числа:

~~1~~ 2 3 ~~4~~ 5 ~~6~~ 7 ~~8~~ ~~9~~ ~~10~~ 11 ~~12~~ 13 ~~14~~ ~~15~~ ~~16~~ 17 ~~18~~ 19 ~~20~~ ~~21~~ ~~22~~ 23 ~~24~~ ~~25~~ ~~26~~ ~~27~~

Используем для решения задачи логический массив p из N элементов. Составляем программу:

```
p[1]:=false;
for  i:=2  to  N  do  p[i]:=true;
 i:=2;
while  int64(i)*i<=N  do  begin            Пока не перешли √N
       if  p[i]  then  begin  j:= int64(i)*i;
                         while  j<=N do  begin
                         p[j]:= false;  j:=j+i;
                         end;
                    end;
       if  i=2 then  i:=3  else  i:=i+2;
end;
for  i:=2  to N  do  if  p[i] then  write  (i, ‘ ‘);
```

Вычеркиваем числа кратные i, начиная от i*i, т.к. остальные имеют делитель меньший чем i и уже вычеркнуты ранее

- В математике доказано, что сложность алгоритма $N*\log_2(\log_2 N)$, значит при $N=10^6$ будет около 4,5 млн. действий и за 1 секунду успеем найти ответы.

3. *Закрепление нового материала* (*репродуктивный метод обучения, индивидуальная форма работы*). Самостоятельная работа в тестирующей системе

Учащимся предлагается зайти на сайт acmp.ru, самостоятельно составить и отослать на проверку в тестирующую систему программу для решения *задачи № 349*.

Задача (№ 349). Найти все простые числа от M до N. ($2 <= M <= N <= 10^6$)

Предлагаю учащимся сделать задачу двумя способами: при помощи функции Prost и при помощи Решета Эратосфена (если осталось мало времени на уроке, то учащиеся делятся на два варианта и реализуют по одному способу).

4. *Подведение итогов урока. Рефлексия*

Выясняем, сколько времени выполнялась программа задачи № 349 каждым из способов.

– Во сколько раз на практике алгоритм Решета Эратосфена оказался быстрее, чем использование функции Prost?

– Какие новые алгоритмы вы освоили? Расскажите основные идеи этих алгоритмов.

5. *Домашнее задание*

1) Повторить алгоритмы: НОД, нахождение делителей, определения простое ли число, алгоритм Решета Эратосфена;
2) на сайте ***acmp.ru*** сдать в тестирующую систему задачу № 60 (Сверхпростое число) и № 171 (Количество делителей).