Перспектива: сборник статей IV Международной научно-практической Интернет-конференции. Вып. 4-2. – Краснояр. гос. пед. ун-т им. В.П. Астафьева. – Красноярск, 2010. – С.140-145.

Использование математических знаний для оптимизации решения олимпиадных задач с программирования

Присяжнюк Т.А.

Житомирский государственный университет имени Ивана Франка

г. Житомир, Житомирская обл., Украина

В статье осветлено основные пути подготовки школьников к участию в олимпиадах с программирования. Показано примеры применения алгоритма Эвклида и функции Эйлера для решения задач, а также их реализация на языке программирования Pascal.

В наше время, когда происходит постоянное обновление информационного общества, все большего внимания научных работников и практиков, привлекает проблема квалифицированной подготовки школьников к олимпиадам с информатики разных уровней, поскольку развитие информационных технологий и стремительный рост объемов информации открывают новые возможности для учеников.

Подготовка к олимпиаде – это длительный трудоемкий процесс, который требует от учеников концентрации внимания, развития логического мышления, а также умения оптимизировать любую задачу. Обязательным условием, при проведении соревнований, является выполнение задач за определенный промежуток времени (то есть лимит

времени ограничен). Для примера наведем сайт www.e-olimp.com, который используется для подготовки и проведения соревнований разных уровней [1, 2]. Для каждой задачи, которая подана к общему списку заданий, устанавливается определенный лимит времени.

Важным аспектом подготовки школьников к соревнованиям являются умения и навыки построения оптимальных алгоритмов решения задач. Одним из перспективных путей оптимизации есть использование математических знаний.

Для заданной задачи ищем инвариантную ей задачу, решение которой будет значительно проще от начальной. Так, например, как известно из математики, исходя из свойств степеней, $a^m \cdot a^n = a^{m+n}$, $a^m : a^n = a^{m-n}$ и так далее, операцию умножения степеней можем заменить сложением показателей, деление степеней, – вычитанием показателей, и тому подобное. Таким образом задача существенно упрощается, соответственно упрощается и ее исследование. Аналогичный подход к анализу задач позволит провести оптимизацию алгоритма ее решения.

Приведем один из методов оптимизации алгоритма на примере решения такой задачи.

Задача. Сколько существует правильных несократимых дробей на промежутке (0;1), знаменатель которых не превышает натуральное число N?

Сначала напомним, какие дроби называются правильными и несократимыми. Дробь, числитель которой меньший знаменателя, называется правильной, а несократимой – числитель и знаменатель которых не имеют общих делителей.

Первый способ решения данной задачи школьниками можно спрогнозировать: они предложат прямой перебор. Очевидно, что все дроби, в которых числитель, – 1, а знаменатель – от 2 к *N*, – будут правильными несократимыми, потому начальное значение количества несократимых дробей будет составлять *N-1*. Для всех других дробей, в которых числитель не равняется 1, будем рассматривать те случаи, в которых знаменатель находится на промежутке от 3 к *N*, а числитель – от 2, но меньший знаменателя. Для каждой пары числитель-знаменатель выясняем: сколько будет общих множителей (не учитывая 1); если такие отсутствуют, то количество несократимых дробей увеличиваем на 1, иначе переходим к следующей дроби.

Примечание: если *N=1*, то понятно, что количество таких дробей будет равно нулю.

Реализуем данный алгоритм решения задачи языком программирования Паскаль.

```
program prjamui_perebir;
var n, i, j, k : int64;
   l, flag : integer;
begin
     read(n);
     if n=1 then k:=0  else
    begin
        k:=n-1; i:=2;
        while i<=n-1 do
```

```
begin
    j:=i+1;
    while j<=n do
    begin
        flag:=0;
      if (i mod 2=0) and (j mod 2=0) then flag:=1
       else
       for l:=3 to trunc(j/2) do
       if (j mod l =0) and (i mod l =0) then
          begin
              flag:=1; break
          end;
      if flag=0 then inc(k);
      inc(j)
    end;
  inc(i)
 end
end;
writeln(k)
end.
```

Данный алгоритм решения является правильным, но превышает лимит времени, поскольку удовлетворяет лишь 52 % тестов, предложенных на вышеуказанном сайте, который мы используем для подготовки и проведения соревнований с информатики разных уровней.

Следующим этапом решения этой задачи является ее оптимизация. Понятно, чтобы упростить наш алгоритм, достаточно для каждой пары числитель-знаменатель найти НОД (наибольший общий делитель) за известным алгоритмом Евклида [3;4], и если он равен 1, то будем увеличивать счетчик количества дробей на 1.

Покажем реализацию предложенного метода на языке программирования Паскаль.

```
program nsd;
var n, i, j, k : int64;
   l, flag : integer;
function nod (a,b:int64):int64;
var x,y :int64;
begin
   x:=a;  y:=b;
   while x<>y do
      if x>y then x:=x-y else y:=y-x;
   nod:=x
end;
begin
```

```
 read(n);
 if n=1 then k:=0 else
begin
   k:=n-1;  i:=2;
  while i<=n-1 do
  begin
       j:=i+1;
      while j<=n do
     begin
        if nod(i,j)=1 then inc(k);
        inc(j)
      end;
      inc(i)
  end
end;
writeln(k)
end.
```

Данное решение является правильным, но оно опять превышает определенный лимит времени условием задачи (60 % тестов, приведенных на сайте www.e-olimp.com [1]). Чтобы решить задачу на все 100 % необходимо обратиться к известной функции Эйлера [5–8], которая позволяет, не перебирая все возможные варианты, для заданного числа N

вычислить количество чисел меньших чем N и взаемопростих с ним. Нахождение функции Эйлера, которая изучается на факультативах с математики, осуществляется следующим образом. Если натуральное число p является простым, то количество чисел взаемопростих с ним будет вычисляться по формуле: $\varphi(p) = p - 1$. Иначе, любое натуральное число можно подать в каноническом виде $p=p_1^{\alpha}p_2^{\beta}...p_n^{\gamma}$, (где $p_1, p_2, \dots p_n$ – простые числа) для которого количество чисел, взаемопростих с данным числом *p*, будет вычисляться по формуле:

$$\varphi(p) = (p_1^{\alpha} - p_1^{\alpha-1})\left(p_2^{\beta} - p_2^{\beta-1}\right) \dots (p_n^{\gamma} - p_n^{\gamma-1})$$

Покажем алгоритм решения задачи с использованием функции Эйлера.

```
var n, i, j, k : longint;
    l, flag : integer;
function ejler (a:longint):longint;
var x,i1,j1,j2,k1,zz,o :longint;
  f1,ej : longint;
begin
    x:=a;  ej:=1;  i1:=2;  j2:=2; k1:=0;
   zz:=trunc(sqrt(x));
   while j2<=zz do
   begin
     if x mod j2 =0 then begin k1:=1; break; end;
```

```
        inc(j2)
      end;
     if k1=0 then ej:=x-1  else
    while x>1 do
    begin
      j1:=0;
     if (x mod i1=0) then
    begin
       while (x mod i1 = 0) do
      begin
         x:=x div i1;  inc(j1)
       end;
       o:=i1;
       for f1:=2 to j1 do
      o:=o*i1;
       ej:=ej*(o-(o div i1))
    end;
   inc( i1)
   end;
   ejler:=ej
end;
```

```
begin
  read(n);
 if n=1 then k:=0 else
 begin
   i:=2;
  while i<=n do
 begin
     k:=k+ejler(i);
     inc(i)
  end
 end;
 writeln(k)
end.
```

Следовательно, как видим, использование некоторых элементов математики способствует значительному упрощению алгоритма решения задачи, а именно его оптимизации по времени. Поэтому при подготовке школьников к олимпиадам с программирования рядом с умением программировать важным есть владение математическими знаниями, а также умение логически совмещать оба компонента.

Литература:

1. www.e-olimp.com

2. Жуковський С.С. “E-olimp” – система автоматичної перевірки задач та проведення олімпіад з інформатики в інтернеті /Комп’ютер у школі та сім’ї. – №1 (65). – 2008. – С.48-50.

3. http://ru.wikipedia.org/wiki/Алгоритм_Евклида

4. http://borlpasc.narod.ru/docym/prac/algorev.htm

5. http://algolist.manual.ru/maths/count_fast/phi_n.php

6. http://e-maxx.ru/algo/euler_function

7. http://dic.academic.ru/dic.nsf/ruwiki/102639

8. http://ru.wikipedia.org/wiki/Функция_Эйлера