

УДК 372.868.1.046.14

А.И. Бочкин, А.С. Слепухина

Анализ некоторых аспектов мыслительной деятельности участников областной олимпиады по информатике

На данном этапе развития образования в Беларуси, как и во всем мире, главная задача школы – гармоническое развитие личности, что должно достигаться и за счет развития творческих способностей. В педагогическом процессе для реализации этой цели применяют различные методы обучения, важнейшими из которых считаются проблемные и исследовательские, и формы обучения, наибольшее значение из которых имеет самостоятельная работа. Во внеучебное время проводятся олимпиады, которые должны выявить учащихся с наиболее развитыми способностями и эрудицией.

На олимпиадах предлагаются довольно сложные задачи, при решении которых учащиеся должны проявить свои способности. Олимпиадные задачи по информатике часто излишне математизируются, что требует от участников хорошей математической подготовки. Но, поскольку цель олимпиады по информатике – “выявление одаренных и образованных школьников” [1] – предлагается альтернативный подход к содержанию задач, заключающийся во введении в не очень сложную задачу некоторых ограничений на средства решения задачи. Это ведет к усложнению поиска алгоритма. Таким образом создается проблемная ситуация, заставляющая искать новые пути решения задачи – проявлять свои способности. Анализ решения таких задач позволяет проследить, хотя бы частично, ход решения задачи и те средства, которые использовались для решения и таким образом выявить школьников “с развитыми способностями к логико-алгоритмическому мышлению...с развитым системно-комбинаторным мышлением” [2].

На областной олимпиаде была предложена одна задача с ограничениями – задача “Игра”. Два первоклассника – Петя и Вова – любят играть в игру со словами: каждый пишет по три слова, затем считает количество совпадающих и несовпадающих букв (совпадающая буква – которая есть в записи каждого из трех слов). Если в записи слова некоторая буква встречается несколько раз, считается она только один раз. В записи слов разрешены только буквы. Выиг-

рывает тот, у кого больше несовпадающих букв. Например, если Петя записал авс,ва,св, а Вова – св, кав, рсв, то выиграл Вова, так как в его словах несовпадающих букв 4(с,к,а,р), а в Петиних словах несовпадающих букв 2(а,с).

Напишите программу, имитирующую такую игру на компьютере. В тексте программы запрещены: 1) глубина вложения циклов более двух; 2) операторы перехода GOTO; 3) все стандартные процедуры и функции, кроме функций вычисления длины текстовой величины и вырезки одного символа. Примечание: вызов из тела цикла, находящегося в основной программе, некоторой процедуры или функции, содержащей более одного вложенного цикла, будет считаться нарушением ограничений.

Указания к решению задачи: Организовать поочередный ввод исходных данных – сначала Петя вводит свои три слова, затем Вова вводит свои три слова. Результат вывести в форме:

Петя <Кол. Совпадающих букв>, <Сами совп. буквы>, <Кол. Несовпадающих букв>, <Сами несовп. буквы>.

Вова <Кол. Совпадающих букв>, <Сами совп. буквы>, <Кол. Несовпадающих букв>. <Сами несовп. буквы>.

Выиграл: <Имя>.

Для приведенного выше примера результат будет такой:

Петя: 1 в 2 а,с;

Вова: 1 в 4 с,к,а,р.

Выиграл: Вова.

Постановка задачи:

Исходные данные – три слова одного игрока и три слова другого игрока.

Результат – сообщения о совпадающих и несовпадающих буквах в словах каждого игрока и сообщение, кто выиграл(пример в условии задачи).

Идея решения:

Надо посчитать количество совпадающих и несовпадающих букв в Петиних словах, в Вовиных словах. Сравнить количество несовпадающих букв у Пети и Вовы и сделать вывод о выигравшем. Поскольку глубина вложения циклов ограничена, нельзя сравнивать буквы всех слов перебором при помощи трех вложенных циклов. Тогда ход решения возможен такой.

1) отсортировать каждое введенное слово (чтобы одинаковые буквы стояли рядом) – авасв ~ ааввс;

2) заменить некоторым символом, например *, повторяющиеся в пределах данного слова буквы – ааввс ~ а*в*с;

3) слить три слова одного игрока в одно слово, три слова другого игрока – в другое слово, выбросив * – а*в*с, ав**с*к, вс*к* ~ авсавсквск;

4) отсортировать полученные слова – авсавсквск ~ аавввссскк,

5) посчитать количество подряд идущих одинаковых букв – если оно равно трем, значит, данная буква есть во всех трех словах, то есть она – повторяющаяся и ее записать в массив повторяющихся букв; если количество меньше трех, то данная буква – не повторяющаяся;

6) вывести на экран результат.

Принцип формирования тестов:

1) надо выявить, выбираются ли вообще одинаковые и неодинаковые буквы в самом простом случае - когда они есть в заданных словах, и когда в самих словах нет повторяющихся букв. Для этого задать слова, в которых встречается одна совпадающая буква, а в самих словах повторяющихся букв нет – abc,ba,cb и cb,dab,rcb.

2) выявить, правильно ли выбираются буквы, когда в одном или нескольких словах есть повторяющиеся буквы – aab,bcb,abc и abb,bdda,aaa.

3) выявить, правильно ли выбираются одинаковые буквы, когда их много и

когда в словах есть много повторяющихся букв (чтобы не было повторяющихся букв при выводе) – aababc, bacc, cbba и baab, cbc, cbb.

4) выявить, правильно ли отработает программа, когда одинаковых букв в словах нет, но попарно в словах одинаковые буквы есть – aab, bcc, dd и aba, bbc, caa .

5) выявить, правильно ли выбираются в результирующем слове совпадающие и несовпадающие буквы, когда первая несовпадающая и последняя несовпадающая – caab, bcb, cd и a, b, cd.

Обычно задачи со сравнением букв в словах решают с использованием вложенных циклов – сколько слов, столько циклов. Но когда слов много, этот путь решения не оптимален. Эффективнее использовать следующую стратегию: 1) из каждого слова удалить повторяющиеся буквы; 2) из всех слов скомпоновать одно; 3) отсортировать его и посчитать количество подряд расположенных одинаковых букв. Если это количество совпадает с количеством слов, то данная буква – совпадающая, в противном случае – несовпадающая. На такой путь решения мы и хотели натолкнуть участников олимпиады, запретив использование более двух вложенных циклов. Запрет на использование стандартных функций должен был исключить применение широких возможностей языка TR7.0 для работы со строками.

Безусловно, описанный путь решения данной задачи – не единственный. Поэтому одно из важнейших требований к участникам – дивергентность мышления, т.к. только предвидя множество путей решения задачи, можно выбрать наиболее оптимальное решение, удовлетворяющее требованиям задачи.

Еще одно требование к участникам – для обхода ограничений надо применить некоторые приемы логико-алгоритмического мышления – конструирующее мышление – в целях удаления символов из слов без использования стандартной функции DEL (надо самостоятельно написать процедуру, выполняющую такое действие), а также прием бисоциативного мышления, чтобы отойти от стандартного способа решения задач этого типа и найти решение, в котором не используется третий вложенный цикл. И те участники, которые решили эту задачу с учетом ограничений, действительно применили указанные приемы мышления.

Решили задачу 9 человек. Из них 5 не нарушили ограничения (Игорь А., Вадим М., Максим П., Павел С., Андрей С.), 2 человека нарушили ограничения незначительно (Андрей Н. – использовал стандартную функцию Delete, Александр К. – использовал различные стандартные функции), 2 человека нарушили ограничения значительно (Андрей Б., Олег Т. – решали задачу стандартно, используя три вложенных цикла).

Для преодоления ограничений Игорь А. организует дополнительный массив для хранения букв, совпадающих в первом и втором словах. Затем эти буквы сравнивает с буквами третьего слова и выбирает совпадающие. Аналогично выбираются несовпадающие буквы. Такой способ обхода ограничений свидетельствует о применении приема заменяющего мышления – три вложенных цикла заменяются последовательностью из двух вложенных циклов. Одновременно используется прием конструирующего мышления для создания массивов совпадающих и несовпадающих букв. Ограничение на использование стандартных функций он обходит путем замены повторяющихся в слове букв пробелом – это результат приема бисоциативного мышления. Такие же приемы мышления использовал Павел С., только по результатам сравнения первых двух слов создает новое слово, а не массив.

Вадим М. также для преодоления ограничений применяет величины других типов – величины типа запись для хранения введенных слов, причем второе поле типа Boolean получает значение true для совпадающих букв. После про-

смотра трех слов совпадающая буква записывается в массив совпадающих букв, несовпадающая – в массив несовпадающих букв. При решении этой задачи ученик применил приемы конструирующего мышления и бисоциации, используя структурированный тип данных и логические величины для преодоления запрета на третий вложенный цикл и используя логические величины в процессе записи совпадающих (несовпадающих) букв в массив – прежде, чем букву записать, проверяется есть ли уже такая буква в массиве. Такие же приемы мышления использовал Максим П. с той лишь разницей, что он не использует величины типа запись, а создает дополнительные массивы – индикаторы повторяющихся букв в слове (на месте повторяющихся букв записывается 1).

Отличается от всех описанных выше программа Андрея Н. Стратегия его решения такова: 1) из трех введенных слов формирует одно(ss); 2) сортирует его; 3) удаляет в нем повторяющиеся буквы; 4) каждую букву этого слова ищет в каждом из трех слов, для чего организует внешний цикл по буквам этого слова и в нем последовательно три цикла по буквам каждого из трех слов. Результат поиска в каждом слове очень оригинален – это логическая величина, которая получает значение true при обнаружении искомой буквы в слове; 5) с помощью составного условия проверяются значения этих трех логических величин – если все три логические величины имеют значение true, буква считается совпадающей, иначе – несовпадающей.

```

for j:=length(ss) downto 1 do
begin
  c:=ss(j);
  f1:=false; f2:=false; f3:=false;
  for i:=1 to length(s1) do if s1[i]=c then begin f1:=true;break end;
  for i:=1 to length(s2) do if s2[i]=c then begin f2:=true;break end;
  for i:=1 to length(s3) do if s3[i]=c then begin f3:=true;break end;
  if f1 and f2 and f3 then
begin
  inc(sc);
  s:=s+c+' ';
end
else begin
  inc(nc);
  n:=n+c+' ';
end;
end;

```

Здесь очень ярко проявилось бисоциативное мышление для преодоления всех ограничений. Программа получилась очень небольшая и понятная.

Таким образом, задачу с ограничениями смогли решить те участники олимпиады, которые применили приемы мышления, позволяющие решать задачи творчески – заменяющее, конструирующее и бисоциативное мышление.

ЛИТЕРАТУРА

1. **Бочкин А.И.** Олимпиадные задачи по информатике // Информатизация адукації, 1996, №2. С. 5-10.
2. **Бочкин А.И., Кунцевич С.П.** Метод ограничения средств при разработке олимпиадных задач // Информатика и образование, 1997, №1. С. 17-20.

S U M M A R Y

The thinking manner of olympiad participant used during solve the increased difficulty task that was generated by the restrictions system, are described.