ИНФОРМАТИКДАН ФАН ОЛИМПИАДАСИ ВИЛОЯТ БОСҚИЧИНИНГ ҚИЗИҚАРЛИ МАСАЛАЛАРИ ТАҲЛИЛИ

О.А.МАМАТҚУЛОВ, А.А.МЕДАТОВ

В статье приведен анализ ошибок, допущенных при решении предложенных участникам олимпиады задач, и указаны пути их устранения.

The article deals with the analysis of mistakes made in the process of solving the matters offered by the participants of the Olympic games and the author offers the ways of liquidating them.

Истеъдодли ва қобилиятли ёшларни аниқлаш ва рағбатлантириш, фан соҳасидаги билимларни тарғиб қилиш ва касбга тайёрлаш информатикадан ўтказилаётган фан олимпиадаларнинг асосий мақсадларидан биридир. Ушбу мақсадларни амалга ошириш қуйидаги вазифаларни амалга ошириш билан боғлиқ бўлади:

- ўқувчи ёшларнинг интеллектуал ривожланишини рағбатлантириш ва қувватлаш;
- истеъдодли ва қобилиятли ёшларни қўллаб-қувватлаш;
- кейинги касбий фаолиятидаги ўз ўрнини аниқлашларига кўмаклашиш;
- информатика фанини ўқитиш сифати ва сама-

радорлигини ошириш;

- иқтидорли ёшлар билан ишлаш методикасини такомиллаштириш;
- ўқитувчиларни информатика соҳасидаги иқтидорли ёшлар билан ишлаш тажрибаларини умумлаштириш ва бошқалар.

Информатикадан ўтказиладиган олимпиада ЭҲМларда турли типдаги масалаларни ечишга қаратилган интеллектуал мусобақадир. Масалани ечиш учун уни чуқур таҳлил қилиш, алгоритмини ва бирор дастурлаш тилида дастурини тузиш, шунингдек, берилган тест бўйича натижани текшириш билан боғлиқ бўлади.

Андижон алоқа коллежида 2012 йилнинг 26-29 март кунлари КҲК ва АЛ талабалари ўртасида информатика фани олимпиадасининг вилояти босқичи бўлиб ўтди.

Олимпиаданинг энг қизиқарли, лекин ечишда деярли барча қатнашчилар турли камчиликларга йўл қўйган масалаларини кўриб чиқамиз.

Деярли барча иккинчи курс қатнашчилари ечишга муваффақ бўлган масалалардан бири иккинчи масала бўлди: *Квадрат илдизи бутун бўлган сонларни кетма-кет (пробелсиз) ёзилса, қуйидагича сатр хосил бўлади: 149162536496481 .... Шу сатрнинг N-ўрнида қандай рақам турганини аниқловчи дастур тузинг.*

Ушбу масалани ечиш кўпчиликка қийинчилик туғдирмади, лекин деярли барча бир-бирига ўхшаш кичик камчиликка йўл қўйгани аниқланди. Қатнашчилар керакли узунликдаги сатрни тузиб олиб, кейин ундан керакли ўриндаги битта рақамни қирқиб олиш йўлини танлаганлар. Бу йўлнинг асосий камчилиги – сатр жуда узун бўлиб, хотирага сиғмай қолади. Шунинг учун сатрни тўлиқ тузишнинг кераги йўқ, фақат унинг узунлиги ҳақидаги маълумотни алоҳида хотирада сақлаб бориш керак эди. Шунда керакли узунликка сатр етиб боргач ,сўралаётган ўрин ва ҳосил бўлган сатр узунлиги орасидаги фарқдан керакли рақамни аниқлаш мумкин бўларди.

Мумкин бўлган тўғри ечимлардан бирининг pascal тилидаги кўринишини қуйида келтирамиз:

```
program P2_2;
var s:string;
   n,i,L,t:longint;
begin
   read(n);
   i:=1; L:=0;
   repeat
     str(i*i,s);
     t:=length(s);
     L:=L+t;
     inc(i);
   until n<=L;
   writeln(s[n+t-L]);
end.
```

Иккинчи курс қатнашчилари ҳал қилишга уринган кейинги (бешинчи) масала: *Иккита қути бор. Биринчи қутида A дона, иккинчисида B дона шарчалар бор ($0<A+B<2^{31}$). Қутиларнинг биридан иккинчисига шарчаларни кўчириш мумкин. Бир қутидан бошқа қутига, бошқа қутида нечта шарча бўлса, шунча шарчаларни кўчириш мумкин. Шундай дастур тузингки, у неча марта кўчириш орқали барча шарчаларни қутилардан бирортасига тўплашнинг мумкинлигини аниқлаб берсин. Агар барча шарчаларни битта қутига тўплашнинг иложи бўлмаса, -1 чиқарилсин.*

Барча қатнашчилар деярли тўғри усул танланганлар: A ёки B лардан бири нолга тенг бўлмагунча такрорлашлар сонини аниқлаш дастурини тузганлар. Барча шарчаларни битта қутига йиғишнинг иложи йўқ бўлган холатни аниқлаш учун навбатдаги A ёки B лардан бирининг дастлабки қийматига тенг эканлигини текширишган.

Кўп ҳолларда бу тўғри, лекин ҳар доим эмас. Масалан, A=1 ва B=19 бўлганда қуйидагига эга бўламиз:

Қадам	1	2	3	4	5	6	7
A	1	2	4	8	16	12	4
B	19	18	16	12	4	8	16

Кўриниб турганидек 5-қадамданоқ 3-қадамдаги қийматлар қайтарилмоқда. Қатнашчиларнинг барчаси шу бир турдаги хусусий ҳолатни ҳисобга олмаган.

Ушбу муаммони хал қилиш учун фақатгина дастлабки қийматлар такрорланаётганлигини текшириш етарли эмас, балки барча ҳосил бўлаётган қийматларни буфер хотирада сақлаб ва текшириб бориш зарур. Буферда A ёки B лардан фақат биттасинигина сақлаш кифоялигининг сабабини уларнинг йиғиндиси қадамлар бўйлаб ўзгармаслиги билан изоҳлаш мумкин.

Мумкин бўлган тўғри ечимлардан бирининг pascal тилидаги кўринишини қуйида келтирамиз:

```
program P2_5;
const MAX_BUFFER_SIZE = 10000;
var a,b,n:integer;
   buffer:array[0..MAX_BUFFER_SIZE] of integer;
   buffer_size:integer;

procedure check(pa,pb:integer);
var i:integer;
begin
 for i:=0 to buffer_size do
      if (buffer[i]=pa) or (buffer[i]=pb) then
```

```
    begin
      writeln(pa:5,pb:6);
     writeln(-1);
     halt;
    end;
    if (buffer_size < MAX_BUFFER_SIZE) then
    begin
       inc(buffer_size);
       buffer[buffer_size]:=pa;
     end;
   end;

   begin
     read(a,b);
     buffer_size:=0;
     buffer[0]:=a;
     n:=0;
     while (a<>0) and (b<>0) do
     begin
        if (a>b) then
          begin a:=a-b; b:=b+b; end
        else
          begin b:=b-a; a:=a+a; end;
     inc(n);
     check(a,b);
     writeln(a:5,b:6);
   end;
   writeln(n);
  end.
```

Учинчи ва тўртинчи масалалар барча қатнашчиларга мушкуллик қилган. Ушбу иккала масала комбинаторикага оидлиги ва энг мақбул вариантни қидириш кераклиги билан бир-бирига ўхшаш.

Улардан соддароғи – тўртинчиси, шунинг учун дастлаб уни кўриб чиқайлик: *Сонлар «занжири» қуйидагича тузилади: занжирдаги биринчи сон ихтиёрий танланади. Иккинчи сондан бошлаб ҳар бир соннинг биринчи рақами ўзидан олдинги соннинг охирги рақамига тенг бўлиши керак. Масалан, 1236, 697, 72, … . Занжирда иккита бир хил сон бўлиши мумкин эмас. Киритилган N та натурал сондан энг узун занжир тузувчи дастур тузинг. Бундай занжирлар бир нечта бўлса, улардан биттасини олиш етарли.*

Ушбу масалани ҳал қилиш учун мумкин бўлган барча комбинацияларни текшириб чиқиш керак бўлади (перебор с возвратом). Бунинг учун ўзаро ичма-ич жойлашган ва сони олдиндан маълум бўлмаган (Nта) цикларни ташкил қилиш керак бўлади. Бу ишни рекурсив функция ёрдамида амалга ошириш қулай. Ҳар бир рекурсив чақирувдан кейин тузилаётган занжирда мумкин бўлган сонларнинг (биринчи ва охирги рақамларидан келиб чиқиб) барчаси қатнашиб бўлгани текшириб кўрилади.

Тузилган биринчи занжир алоҳида ҳотирада сақланиб, вақтинча энг узун занжир деб ҳисобланади. Ҳар кейинги узунроқ занжир унинг ўрнини эгаллайди.

Мумкин бўлган тўғри ечимлардан бирининг pascal тилидаги кўринишини қуйида келтирамиз:

```
program P2_4;
const MAX_COUNT = 100;
var n,i,best_length:integer;
  m,curr,best:array[1..MAX_COUNT] of integer;
  used:array[1..MAX_COUNT] of boolean;

function first(x:integer):char;
var s:string;
begin
  str(x,s);
  first:=s[1];
end;

function last(x:integer):char;
var s:string;
begin
  str(x,s);
  last:=s[length(s)];
end;

procedure save(p:integer);
var i:integer;
begin
  if (p > best_length) then
  begin
     best_length := p;
     for i:=1 to best_length do
       best[i] := curr[i];
  end;
end;

procedure put(p:integer);
var i:integer;
begin
   if (p>n) then
      exit;

 for i:=1 to n do
    if (not used[i]) and ((p=1) or
      (last(curr[p-1])=first(m[i]))) then
    begin
       used[i]:=true;
       curr[p] := m[i];
       save(p);
       put(p+1);
       used[i]:=false;
```

```
    end;
  end;

  begin
    read(n);
    for i:=1 to n do
    begin
      read(m[i]);
      used[i]:=false;
   end;

   best_length:=0;
   put(1);

   writeln(best_length);

   for i:=1 to best_length do
     write(best[i],' ');
   writeln;
  end.
```

Иккинчи курс қатнашчилари учун қўйилган масалалар ичида энг мушкули бўлган учинчи масалага ўтсак: *N ($N<10^3$) натурал сон ва A[1,N] камаймайдиган тартибда сараланган натурал сонлар массиви берилган ($A[i]<10^2$). Берилган S сонни энг кам сондаги A[1], A[2], …, A[N] элементлар йиғиндиси ёрдамида ифодаловчи дастур тузинг. Бунинг имкони бўлмаса, шу ҳақида хабар чиқарилсин. Массив элементларини бир неча марта ишлатиш мумкин.*

Бу масала ҳам худди тўртинчи масалага ўхшаш ечимга келади. Фарқи кўриладиган ҳар бир элемент бир неча мартадан қатнашиши мумкинлигида. Яна иложи борича кам сондаги элементлардан фойдаланишга ҳаракат қилиниши керак. Бу ечим учун номзод бўлган комбинациялар сонини тез қисқартиришга сабаб бўлади. Чунки биринчи топилган вариант вақтинча энг яхши деб сақлаб турилади. Кейинги вариантлар кўрилаётганда элементлар сони ортиб кетиши билан ўша вариантни кўришни давом эттиришни тўхтатиб навбатдаги вариантни кўришга ўтилаверади.

Мумкин бўлган тўғри ечимлардан бирининг pascal тилидаги кўринишини қуйида келтирамиз:

```
program P2_3;
const MAX_COUNT = 1000;
var n,i,s,best_length:longint;
   a:array[1..MAX_COUNT] of integer;
   curr,best:array[1..MAX_COUNT,1..2] of integer;
   used:array[1..MAX_COUNT] of boolean;

procedure save(p:integer);
var i:integer;
begin
  best_length := p;
  for i:=1 to best_length do
    best[i] := curr[i];
end;

procedure put(p,s:integer);
var i,j:integer;
begin
   if (p>best_length) then exit;
   if (s=0) then
   begin
     save(p-1);
     exit;
  end;
  for i:=n downto 1 do
    if (not used[i]) then
    begin
      used[i] := true;
      curr[p][1] := i;
      for j:=(s div a[i]) downto 1 do
      begin
        curr[p][2] := j;
        put( p+1, s - j * a[i] );
      end;
     used[i] := false;
   end;
end;


begin
  read(n);
  for i:=1 to n do
  begin
    read(a[i]);
    used[i] := false;
  end;
  read(s);

  best_length := n+1;
  put(1,s);

  if (best_length < n+1) then
  begin
    writeln(best_length);
    for i:=1 to best_length do
    begin
      if (i>1) then write(' + ');
      write(best[i][2],'*',a[best[i][1]]);
    end;
    writeln;
  end
  else
     writeln(-1);
end.
```

Хулоса қилиб айтганда, олимпиада юқори са-

вияда ташкил этилди. Ўқувчи ёшлар олимпиадага катта қизиқиш билан қатнашдилар. Жумладан, 2-курслардан 18 нафар иштирокчилар ўз билимларини синаш учун фаол иштирок этдилар.

Иштирокчиларга ҳал қилиш учун тақдим этилган муаммолар олимпиада савиясига муносиб, мураккаб тузилган эди. Айниқса, хусусий ҳолатлар, комбинаторика масалалари бўйича кўпроқ машқ қилиниши кераклиги аниқланди.

Информатикадан фан олимпиадаларга тайёрлашда нафақат информатика фани ўқитувчилари, балки математика ва бошқа фан ўқитувчилари ҳам яқиндан ёрдам бериб, ўқувчини комплекс тайёрлаш кераклиги мақсадга мувофиқ деб ўйлаймиз. Чунки қўйилган масалаларни ҳал қилишда ўқувчилардан бутун сонли арифметика, кетма-кетликлар, комбинаторика элементлари, векторлар ва матрицалар устида амаллар, сонли усуллар, максимум ва минимумга оид тушунчалар, геометрия ва математик мантиқ элементлари ва бошқалар бўйича чуқур билимга эга бўлиш талаб қилинади. Шунинг учун ўқитувчилардан иқтидорли ёшларни эрта аниқлаб, улар билан мунтазам шуғулланиш натижасидагина ўқувчи ёшларни жаҳон олимпиадаларига олиб чиқишдек масъулиятли ишга эришишимиз мумкин деб ўйлаймиз.

Ушбу мақолада бўлиб ўтган олимпиадада қатнашчилар томонидан масалаларни хал қилишда йўл қўйилган камчиликлар ва уларни бартараф қилиш бўйича кўрсатмалар беришга ҳаракат қилдик. Иқтидорли ёшларимизга кўрсатилган ечимлар шу каби масалаларни ҳал қилишларида яқиндан ёрдам беради деган умиддамиз ва улардан кўрсатилган масалаларнинг бошқа усулдаги ечимларини ҳам топишларига умид билдирамиз.

Адабиётлар:

1. Азларов Т.Р., Аширов Ш.Ш. Информатикадан олимпиада масалаларини ечиш. Мактаб ва ўрта ўқув юртлари ўқитувчилари учун методик қўлланма.-Т.,1993
2. Кирюхин В. М., Окулов С. М. Методика решения задач по информатике. Международные олимпиады. М.: Бином, 2007
3. Медатов А.А. Информатикадан олимпиада масалалари. Методик қўлланма.- Андижон: АДУ, 2011
4. Дастурлаш бўйича билимларни баҳолаш ва интелектуал муносабатларни ташкил этиш корпоратив тизими - http://olymp.tuit.uz/
5. Дастурлаш буйича жахон чемпионати сайти – http://icpc.baylor.edu/
6. Саратов давлат университетининг дастурлаш буйича олимпиада утказиш онлайн тизими сайти - http://acm.sgu.ru/
7. Информатика буйича олимпиадаларга багишланган сайт – http://www.informatics.ru/
8. Россиядаги дастурлаш буйича энг йирик масалалар туплами сайти - http://acm.timus.ru/

Андижон давлат университети,
2012 йил 20 апрелда қабул қилинган
(Тақризчи: проф. А.Муллажонов)