

# ПЪРВА МЕЖДУНАРОДНА ОЛИМПИАДА ПО ИНФОРМАТИКА

ПАВЕЛ АЗЪЛОВ — София

## 1. ОРГАНИЗАЦИЯ И ПРОВЕЖДАНЕ НА ОЛИМПИАДАТА

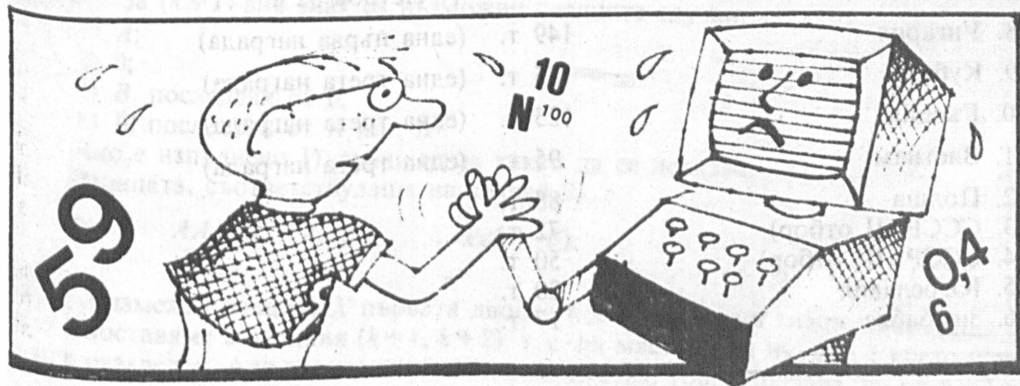
По предложение на акад. Благовест Сендов, представено през 1987 г. на сесия на ЮНЕСКО, от 16 до 19 май тази година в Плевен се проведе Първата международна олимпиада по информатика за средношколци. Организатори на олимпиадата бяха ЮНЕСКО, а от българска страна Министерството на културата, науката и просветата, Съюзът на математиците в България, Институтът по математика при БАН и техникумът по микропроцесорна техника в Плевен. В олимпиадата участваха 16 отбора от 13 страни: Виетнам, ГДР, Гърция, Зимбабве, Китай, Куба, Полша, СССР (три отбора), Унгария, ФРГ, Чехословакия, Югославия и България (два отбора). Отборите бяха с по трима участници, с изключение на Унгария и Югославия, които представиха отбори с по двама ученици. Международното жури с председател чл.-кор. Петър Кендеров състави тема от три задачи на базата на предварително изготвени задачи от страните-участнички. Състезанието се състоя на 17 май от 14 до 18 часа. Всеки ученик имаше на разположение микрокомпютър Плевен—16, Плевен—82 или използваше собствен компютър. Максималната оценка общо за трите задачи беше 100 точки. Точките, които журито определи за отделните награди, бяха съответно:

Първа награда — от 91 до 100 точки;

Втора награда — от 81 до 90 точки;

Трета награда — от 60 до 80 точки.

Наградите се обявиха на официалното закриване и бяха връчени от др. Владимир Живков — председател на фондацията „Л. Живкова“. Всички ученици получиха диплом за участие.



## 2. КЛАСИРАНЕ НА ОТБОРИТЕ

През последните три години нашата страна беше домакин, организатор и участник в три международни състезания по информатика за средношколци:

— Отворено състезание по програмиране, 17—19 май 1987 г., София. участвуват 6 страни:

— „Микрокомпютърът — програмиране и експлоатация“ — международен конкурс за ученици от професионално-техническите училища на социалистическите страни, 5—8 октомври 1988 г., Варна, участвуват 6 страни:

— Първа международна олимпиада по информатика, 16—19 май 1989 г., Плевен, участвуват 13 страни.

Победител във всеки един от тези конкурси е бил винаги български ученик. И представянето на целия отбор — отлично. Това не са случайни факти. У нас има много добри условия за извънкласна работа по информатика. Преди всичко това са широкодостъпната микрокомпютърна техника, системната целогодишна извънкласна работа и разнообразните национални конкурси: трите кръга на националната олимпиада, зимните математически празници на младите математици, състезанието по информатика за ученици от техникумите, което се провежда в Плевен, турнирът по информатика в Пловдив, конкурсите по информатика на списанията „Математика“, „Обучението по математика и информатика“ и „Компютър за Вас“ и други регионални конкурси и състезания.

Класирането на отборите в Първата международна олимпиада по информатика е следното:

1. България (I отбор)	275 т.	Е. Тодоров 95 т. I награда А. Алтънов 90 т. II награда И. Маринов 90 т. II награда
2. Китай	221 т.	(три трети награди)
3. ФРГ	215 т.	(една първа и една втора награда)
4. Чехословакия	209 т.	(една първа награда)
5. ГДР	207 т.	(две втори награди)
6. СССР (I отбор)	190 т.	(една първа и една втора награда)
7. България (II отбор)	188 т.	Т. Тончев 100 т. I награда С. Бонев 56 т. Т. Петров 32 т.
8. Унгария	149 т.	(една първа награда)
9. Куба	135 т.	(една трета награда)
10. Гърция	125 т.	(една трета награда)
11. Виетнам	95 т.	(една трета награда)
12. Полша	88 т.	
13. СССР (II отбор)	72 т.	
14. СССР (III отбор)	50 т.	
15. Югославия	30 т.	
16. Зимбабве	17 т.	

### 3. ТЕМА

Дадени са  $2n$  ( $n > 1$ ) кутии, разположени последователно една след друга, от които две съседни са празни, а в останалите се съдържат  $n-1$  знака  $A$  и  $n-1$  знака  $B$ , като във всяка кутия се съдържа точно по един знак.

Пример. При  $n=5$ : 

A	B	B	A			A	B	A	B
---	---	---	---	--	--	---	---	---	---

**Правило за разместване.** Съдържанието на кои да е две съседни кутии може да бъде преместено в празните кутии, като се запази редът им.

**Цел.** Да се получи редица, в която всички знаци  $A$  се намират отляво на (предхождат) знаците  $B$ , без значение къде са разположени празните кутии.

**Задача.** Да се състави програма, която:

1. Въвежда от клавиатурата началното състояние като редица, елементите на която са  $A$ ,  $B$  и  $0$  (за празните кутии) и моделира разместванията;

2. За дадено начално състояние намира поне една последователност от размествания, която води до целта, или извежда съобщение, че такава последователност не съществува. Изходът трябва да съдържа началното състояние, междинните състояния след всяко разместване и крайното състояние;

3. Намира последователност, водеща до целта с най-малко размествания.

**Бележка.** Да се представи поне едно решение на примера, посочен по-горе.

Ще разгледаме решението на българския ученик Емануил Тодоров от Пловдив. Той получи първа награда с 95 точки. За разлика от всички останали неговото решение съдържа конструктивно доказателство за съществуване на решение при определени условия. Точките, които журито справедливо му отне, бяха заради непълнота в контрола на входните данни.

#### Анализ на задачата

При  $n=2$  има две възможности:

1) знаците са подредени — например  $00AB$ ;

2) знаците не са подредени и не могат да се подредят — например  $BA00$ .

При  $n=3$  съществува начално положение, от което може да се получи последователност от размествания, водеща до целта — например  $BB00AA$ , но има и такова начално положение — например  $00ABAB$ , от което не може да се стигне до целта.

При  $n > 3$  ще докажем, че винаги съществува решение.

Нека в първите  $k$  ( $0 \leq k < n-1$ ) кутии се съдържа знакът  $A$ , т. е. налице е редицата

$$\underbrace{AA \dots A}_{k \text{ пъти}} XX \dots X, \quad X \in \{0, A, B\}.$$

Ще покажем как в  $(k+1)$ -вата кутия може да се постави следващото  $A$  (ако има такава). За  $(k+1)$ -вия знак са възможни следните случаи:

1)  $A$ ;

2)  $0$ ;

3)  $B$ , последван от  $0$ ;

4)  $B$ , последван от  $A$  или  $B$ .

Ако е изпълнено 1), значи няма какво да се доказва.

Редицата, съответстваща на случай 2), е

$$\underbrace{AA \dots A}_{k \text{ пъти}} A00XX \dots X, \quad X \in \{A, B\}.$$

Търсим измежду знаците  $X$  първата двойка последователни знаци, започваща с  $A$ , и я поставяме в позиция  $(k+1, k+2)$ , т. е. на мястото на нулите, с което случаят е разгледан. Ако такава двойка не съществува, това означава, че  $k=n-2$  и

последният знак в редицата е  $A$ , т. е. тя изглежда така:

$$\underbrace{AA \dots A00}_{n-2 \text{ пъти}} \underbrace{BB \dots BA}_{n-1 \text{ пъти}}$$

Ето как може да се продължи по-нататък (двойките знаци, които се разменят на всяка стъпка, са подчертани):

$AA \dots A00 \underline{BB} \dots \underline{BA}$   
 $AA \dots A \underline{B} \underline{A} \underline{B} \underline{B} \dots B00$   
 $AA \dots A \underline{B} \underline{0} \underline{0} \underline{B} \underline{B} \dots \underline{B} \underline{A} \underline{B}$   
 $AA \dots A \underline{B} \underline{B} \underline{B} \underline{0} \underline{0} \underline{B} \dots \underline{B} \underline{A} \underline{B}$   
 $AA \dots A \underline{0} \underline{0} \underline{B} \underline{B} \underline{B} \dots \underline{B} \underline{A} \underline{B}$   
 $AA \dots A \underline{A} \underline{B} \underline{B} \dots B00.$

С това разглеждането на случай 2) е завършено.

Останалите два случая се свеждат към него. Ако редицата е

 $AA \dots A \underline{B} \underline{0} \underline{0} \underline{X} \underline{X} \dots X,$ 

извършваме преобразуванията

$AA \dots A \underline{B} \underline{0} \underline{0} \underline{X} \underline{X} \dots X$   
 $AA \dots A \underline{B} \underline{X} \underline{X} \underline{0} \underline{0} \underline{X} \dots X$   
 $AA \dots A \underline{0} \underline{0} \underline{X} \underline{B} \underline{X} \underline{X} \dots X,$

а ако редицата е

$AA \dots A \underline{B} \underline{A} \underline{X} \underline{X} \dots X$  или  
 $AA \dots A \underline{B} \underline{B} \underline{X} \underline{X} \dots X,$

с едно разместване получаваме

 $AA \dots A \underline{0} \underline{0} \underline{X} \underline{X} \dots X.$ 

Получаването на търсената редица на знаците по изложения алгоритъм изисква  $pn$  премествания, където  $p$  е константа, независеща от  $n$ , т. е. алгоритъмът е с линейна сложност.

Да означим с  $m_2$  броя на разместванията, с които по описания алгоритъм се стига до целта. В общия случай  $m_2$  не е минималното число с това свойство. За да се намери последователността от най-малко размествания, която подрежда знаците, може да се генерират всички възможни решения и от тях да се вземе това, което има най-малък брой ходове, като за целта се използва методът „търсене с връщане назад“.

За да се намали броят на претърсванията, като критерий за минималност може да се постави условието

$$q \leq \min(m_2, m_3),$$

където  $q$  е броят на ходовете в текущото решение, а  $m_3$  е броят на ходовете в най-доброто намерено до момента решение.

Следва текстът на програмата.

PROGRAM PROBLEM;

Var box, box1 : array[1..1000] of char;  
 st, stt : array[1..1000] of byte;  
 spn, n, max : integer;  
 em, em1, mm : integer;  
 flag : boolean;

PROCEDURE INPUT;

var i : integer;  
 begin  
 em := 0;  
 clrscr;  
 write('n = '); readln(n); writeln;  
 for i := 1 to 2\*n do  
 begin  
 write('box ', i, ' ');  
 readln(box[i]);  
 box1[i] := box[i];  
 if (box[i]='0') and (em=0)  
 then em := i  
 end;  
 em1 := em  
 end;

FUNCTION CHECK : boolean;

var i : integer;  
 lst : char; fl : boolean;  
 begin  
 fl := true;  
 lst := box1[1];  
 for i := 2 to 2\*n do  
 begin  
 if (box1[i]='a') and (lst='b')  
 then fl := false;  
 if box1[i]>'0'  
 then lst := box1[i]  
 end;  
 check := fl  
 end;

PROCEDURE PRINT;

var i : integer;  
 begin  
 for i := 1 to 2\*n do  
 write(box1[i], ' ');  
 writeln  
 end;

PROCEDURE MOVE (pos : integer);

begin  
 spn := spn + 1;  
 box1[em1] := box1[pos];  
 box1[em1 + 1] := box1[pos + 1];  
 box1[pos] := '0';  
 box1[pos + 1] := '0';  
 em1 := pos; print  
 end;

```

PROCEDURE MOVE1 (pos : integer);
begin
  spn := spn + 1;
  box1[em1] := box1[pos];
  box1[em1 + 1] := box1[pos + 1];
  box1[pos] := '0'; box1[pos + 1] := '0';
  em1 := pos; st[spn] := pos
end;

```

```

PROCEDURE FINDWAY;
var k,t : integer;
    flag : boolean;
begin
  spn := 0;
  while (not
    begin
      if box1[
        then
          k := k+1
        else
          begin
            if box1[k+1]=
              then
                if box1[k+2]<
                  then move(k+1)
                  else begin
                      move(em1+2);
                      move(k+1)
                    end;
                t := k+1;
                repeat
                  t := t+1
                until (box1[t]='a');
                if t<2*n
                  then move(t)
                  else begin
                      move(t-1);
                      move(k+2);
                      move(k+4);
                      move(k+1);
                      if (not check)
                        then move(2*n-1)
                      end;
                    k := k+1
                  end
                end
            end
          end;

```

```

end;
PROCEDURE BACK;
var i,j : integer;
begin
  em1 := em;
  for i:=1 to 2*n do box1[i] := box[i];
  j := spn-1; spn := 0;
  if j=0 then flag := false
  else begin
    for i:= 1 to j do
      movel(st[i])
    end
  end
end;

```

```

PROCEDURE FORWRD;
var i,t : integer;
begin
  if spn<mm
    then begin
      t := 1;
      while (box1[t]='0') or
        (box1[t+1]='0') do t:=t+1;
      movel(t)
    end
  else begin
    repeat
      t := st[spn]; back;
    repeat
      t := t+1
    until (t=2*n) or
      ((box1[t]<>'0') and
        (box1[t+1]<>'0'));
    if t<2*n then flag := true
    until (t<2*n) or (not flag);
    if t<2*n then movel(t)
  end
end;

```

```

end;
PROCEDURE FINDMIN;
var i : integer;
begin
  if check
    then begin
      mm := spn; em1 := em;
      for i := 1 to 2*n do
        box1[i] := box[i];
      max:=mm+1; spn:=0; flag:=true;
      while (flag) and (spn<=mm) do
        if check
          then begin
              if max>spn then
                begin
                  max := spn;
                  for i:=1 to max do
                    stt[i]:=st[i];
                  mm := max
                end;
              forwrd
            end
          else forwrd;
        writeln; writeln(max);
        for i := 1 to 2*n do
          box1[i] := box[i];
        em1 := em; print;
        for i := 1 to max do
          move(stt[i]);
        end
      else writeln('no way')
    end;

```

```

BEGIN
  input; print;
  findway; findmin
END.

```