I^st International Tournament
of Young Mathematicians
27th June – 3rd July 2009, Paris, France

# PROBLEM SIX

## PATTERN GRAPHS

### General formulation

Let n be a positive integer. A pattern of length n is a two-line table

$$\begin{matrix} a_1\ a_2\ \ldots\ a_n \\ b_1\ b_2\ \ldots\ b_n \end{matrix}$$

Where $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$ are some rearrangements of the numbers $1, 2, \ldots, n$.

Define two operations A and B on patterns as follows
**A :** replace each number a of the first line with the number that is in the a'th place (from left to right) of the second line,
**B :** replace each number b of the second line with the number that is in the b'th place
(from left to right) of the first line.
We can construct an oriented labelled graph $G_n$ whose vertices are all the patterns of length $n$, and such that for any two vertices v and w there is an A-arrow (resp. a B-arrow) from v to w if the pattern w is obtained from the pattern v by applying the operation A (resp. the operation B).

**TEAM:** MATH HIGH SCHOOL *NANCHO POPOVICH,* BULGARIA

**Leaders:** LOZANOV Chavdar, CHRISTOVA Madlen
**Contestants:** VALKOV Mladen, YORDANOVA Yoana, MARKOVA Magdalena, KOSTADINOVA Georgina, ALEKSANDROVA Polina, ENCHEV Ivaylo

## Theoretical part

- Pattern plate is a vertex of the graph, which includes a permutation $\begin{matrix} a_1\ a_2\ \dots\ a_n \\ b_1\ b_2\ \dots\ b_n \end{matrix}$ .
- *A component* of the graph $G_n$ is a connected subgraph, where every edge corresponds to an operation (*A* or *B*).
- *An endmost point of a component* is a vertex of the figure, which is formed (in the segment – the two ends, in the square – its vertex).

## Examination

# Part one

**Problem 1.** Find the number of the vertexes in graph $G_3$.

Solution.

When $n = 3$, the graph has 7 components.

The number of the vertexes in the graph is 36, which is the number of all the permutations of a pattern $\begin{matrix} a_1\ a_2\ a_3 \\ b_1\ b_2\ b_3 \end{matrix}$ - $((3!)^2)$.

| | | | | | |
|---|---|---|---|---|---|
| 123<br>213 | 123<br>231 | 123<br>123 | 123<br>132 | 123<br>321 | 123<br>312 |
| 132<br>213 | 132<br>231 | 132<br>123 | 132<br>132 | 132<br>321 | 132<br>312 |
| 213<br>213 | 213<br>231 | 213<br>123 | 213<br>132 | 213<br>321 | 213<br>312 |
| 231<br>213 | 231<br>231 | 231<br>123 | 231<br>132 | 231<br>321 | 231<br>312 |
| 312<br>213 | 312<br>231 | 312<br>123 | 312<br>132 | 312<br>321 | 312<br>312 |
| 321<br>213 | 321<br>231 | 321<br>123 | 321<br>132 | 321<br>321 | 321<br>312 |

Fig. 1

**Problem 2.** Find the number of the vertexes of the graph $G_n$.

Solution.

We will prove that every permutation of the numbers from 1 to n takes part in the graph $G_n$ only once (every two patterns are different, namely the graph includes $(n!)^2$ vertexes).

Let a fixed pattern takes part in the graph twice.

Let this pattern takes part twice in one component. Then, when we do the operations A and B, we get new patterns (in an opposite case the component consists of only one pattern).

If this element can be found for a second time, after applying operations A and B, the result will be the same as at the first pattern. Then we get two similar components. In the same logic, if we have two similar patterns in two components of the graph, the components are similar.

*Conclusion.*

Every permutation of the pattern $\begin{matrix} a_1\ a_2\ \dots\ a_n \\ b_1\ b_2\ \dots\ b_n \end{matrix}$ occurs only once in the graph $G_n$.

The number of vertexes in the graph $G_n$ is equal to the number of the permutations of the pattern – $(n!)^2$.

**Problem 3.** Examine the components of the graph $G_3$.

Solution.

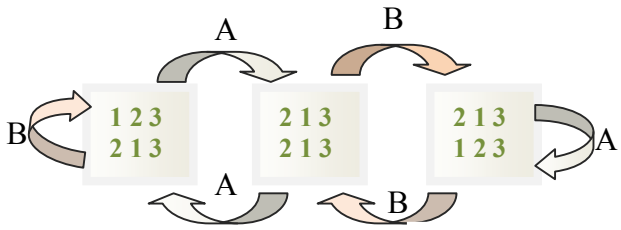



Fig. 2

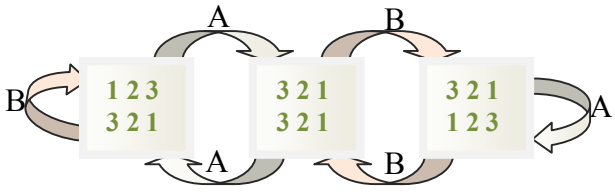



Fig. 3

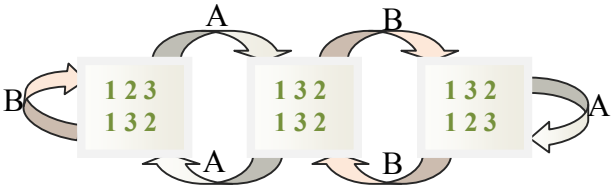



Fig. 4

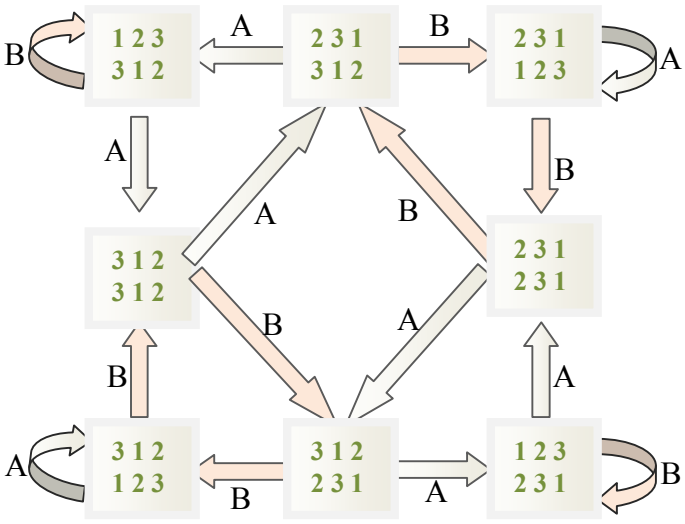



Fig. 5


Fig. 6

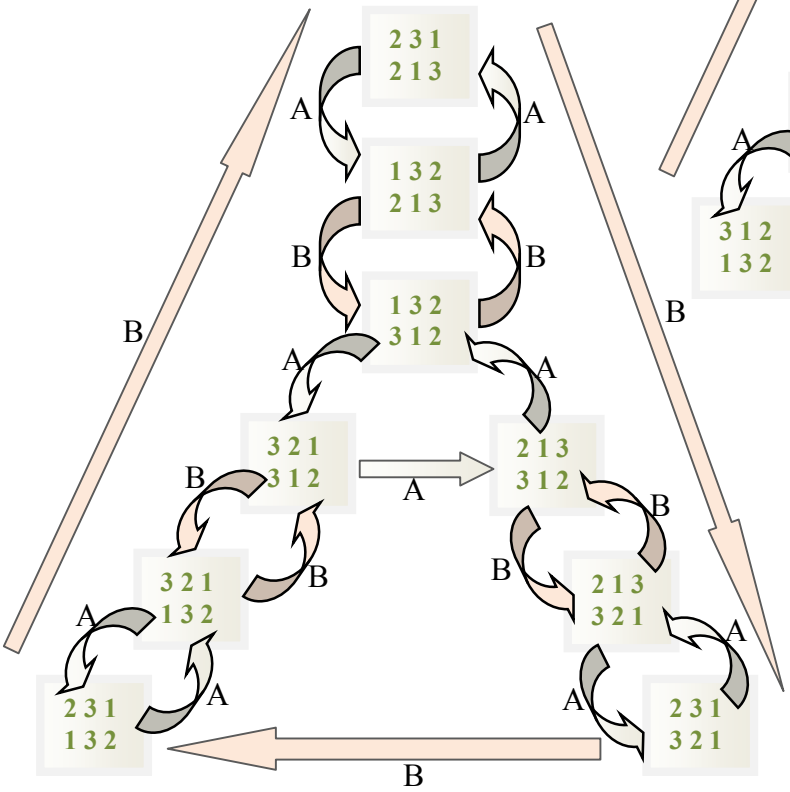



Fig. 8

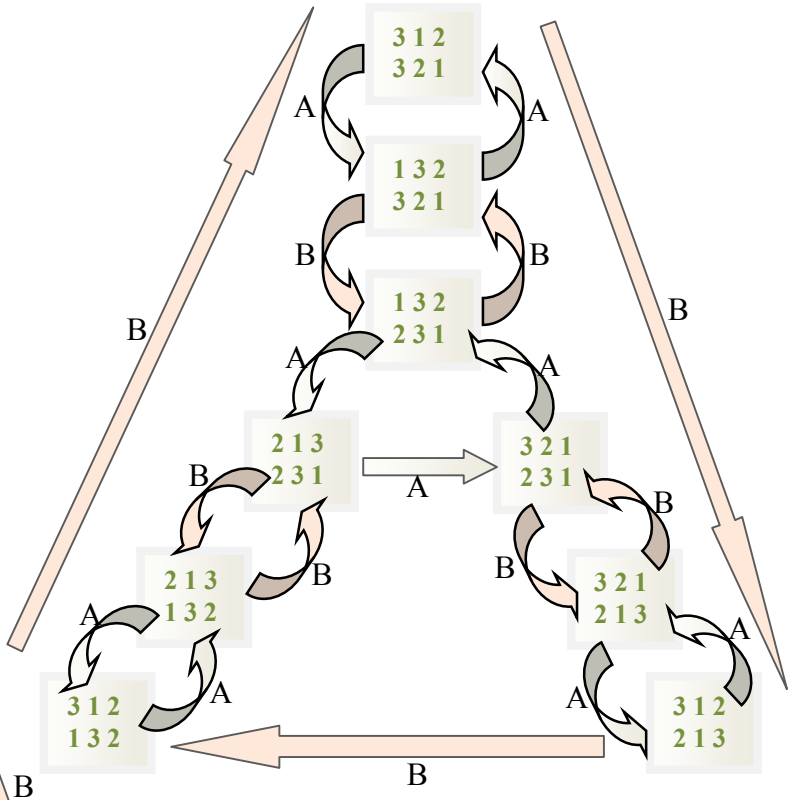



Fig. 7

These are the all components of the graph G3. It has 36 patterns, which make the following components:

- 1 component with 1 pattern;
- 3 components with 3 patterns;
- 1 component with 8 patterns;
- 2 components with 9 patterns.

The number of all components is 7.

**Problem 4.** Examine some geometric properties of the components in the graph G3.

Solution.

All the elements of the graph can be arranged in the plane, so the graph is planar. Every component forms connected graph. In the graph G3, as we can see on the figures, there is no component, whose edges are crossed.
The components with 9 patterns are simple, and all the other are not simple, since they have loops.
The component with 8 patterns is centro-symmetric, and the other components are symmetric.

**Problem 5.** If the connections between every two patterns are regarded as one edge of the subgraph (the component of the graph), we shall prove that:
If the component has $N$ patterns, $M$ edges, and they divide the plane in $F$ areas, and the subgraph is planar, then:

$$N - M + F = 2.$$

Solution.
We will do induction for F. If $F = 1$, then the subgraph (component) is tree and the statement is true. Let $F > 1$, then the statement is true for components, which are planar graphs with less than F areas. When $F>1$ it follows that in the subgraph exists at least one cycle. Let $(u, v)$ is an edge of the component, which belongs to the cycle.
Let J be the graph, which is obtained from the component by removing the vertex $(u, v)$. It is clear that J is connected planar graph, which has $N$ vertexes, $M$–1 edges and $F$–1 areas. Then according to the induction proposition about $J$, the equality

$$N - (M - 1) + (F - 1) = 2$$

is true and it is equivalent to the statement.

**Consequence Problem.** If the graph is planar with $N \geq 3$ vertexes, then it has at most $3N - 6$ edges.

**Problem 6.** Examine the geometrical properties of the components in graph $G_n$.
Solution.
All the components are planar graphs because every pattern can be connected to the other, in that way to have no intersects of the operations. This is because A and B connect consecutive patterns.

**Problem 7.** Examine some properties of the pattern with permutation 1 2 3…$n$.
Solution.
A pattern with the permutation 1 2 3... ($n$-1) $n$.
When a fixed component has a pattern, consisted of the consecutive numbers from 1 to n, then if this permutation is on the first line, after applying operation $B$, the pattern is the same. If the permutation is on the second line, after operation $A$ the pattern is the same. If the permutation is in the both lines – the

component is made of only one pattern and after applying the two operations it isn't changed.

**Problem 8.** Can the pattern $\begin{matrix} 2\ 1\ 3\ 4\ \ldots\ n\text{-}1\ n \\ 2\ 3\ 4\ 5\ \ldots\ n\ \ 1 \end{matrix}$ be obtained from the pattern $\begin{matrix} 2\ 3\ 1\ 4\ \ldots\ n\text{-}1\ n \\ 2\ 3\ 4\ 5\ \ldots\ n\ \ 1 \end{matrix}$ using the operations A and B?

Solution.

The two patterns have same second lines. For keeping the second line we should use only operation A. Then the first line is changing. But when we use only operation A several times, the line returns to its original mode. So it is impossible to obtain the first pattern from the second.

There is the model in which way the first line of the pattern is changing:

*(2 1 3 4 ... n-1 n)→(3 2 4 5 ... n 1)→(4 3 5 6 ... 1 2)→(5 4 6 7 ... 2 3)→... →(2 1 3 4 ... n-1 n).*

The solution is checked by the computer programme, which is offered in problem 9.

**Problem 9.**

Make a computer program, which discover if one pattern can be obtained from other after bringing in them and give some examples.

Solution.

In this programme, you have to lead in two patterns and then the programme will show you if you can obtain the second from the first.

```
#include<cstdio>
#include<vector>
#include<queue>
#include<set>
using namespace std;
struct node
{
 int a[1001];
 int b[1001];
};
int n;
class cmp
{
 public:
  bool operator () (node x,node y)
   {
    int i;
    for(i=0;i<n;i++)
     if(x.b[i]!=y.b[i])
      return x.b[i]<y.b[i];
      for(i=0;i<n;i++)
       if(x.a[i]!=y.a[i])
        return x.a[i]<y.a[i];
      return 0;
    }
};
set <node,cmp> s;
node st;
node t;
int bfs(node f)
{
 queue <node> q;
 q.push(f);
 s.insert(f);
 int br=0;
 while(!q.empty())
  {
   node temp=q.front();
   q
.pop();

    br++;
    int i;
  //  for(i=0;i<n;i++)
  //   printf("%d ",temp.a[i]);
   //  printf("\n");
 //   for(i=0;i<n;i++)
  //   printf("%d ",temp.b[i]);
   //  printf("\n");
   //  printf("\n");
    int l=0;
    for(i=0;i<n;i++)
     if(temp.a[i]!=st.a[i])
      { l=1;break;}
    if(l==0)
     for(i=0;i<n;i++)
      if(temp.b[i]!=st.b[i])
       {
        l=1;break;
```

```
    }
  if(l==0)return 1;
  node A;
  for(i=0;i<n;i++)
   {
    A.a[i]=temp.b[temp.a[i]-
1];
    A.b[i]=temp.b[i];
   }
  if(s.find(A)==s.end())
   {
    s.insert(A);
    q.push(A);
   }
  node B;
  for(i=0;i<n;i++)
   {
    B.b[i]=temp.a[temp.b[i]-
1];
    B.a[i]=temp.a[i];
   }
  if(s.find(B)==s.end())
   {
    s.insert(B);
    q.push(B);
   }
  if(br>100000)return 2;
 }
// printf("no\n");
 return 0;
}
int main()
{printf("Enter n:");
 scanf("%d",&n);
 int i;
 printf("Enter start
node:\n");
 for(i=0;i<n;i++)
  scanf("%d",&t.a[i]);
 for(i=0;i<n;i++)
  scanf("%d",&t.b[i]);
 printf("Enter target
node:\n");
 for(i=0;i<n;i++)
  scanf("%d",&st.a[i]);
 for(i=0;i<n;i++)
  scanf("%d",&st.b[i]);
 int k=bfs(t);
 if(k==0)
  printf("No route found\n");
 if(k==1)
  printf("Route found\n");
 if(k==2)
  printf("No route found in
100000 operations");
 return 0;
}
```

**Addition programme.** This is another programme, which goes round the graph (pattern) and searches for the other vertex with less than 10000 operations.

```
#include<cstdio>
#include<vector>
#include<queue>
#include<set>
using namespace std;
struct node
{
 int a[1001];
 int b[1001];
};
int n;
class cmp
{
 public:
  bool operator () (node
x,node y)
  {
   int i;
   for(i=0;i<n;i++)
    if(x.b[i]!=y.b[i])
     return x.b[i]<y.b[i];
   for(i=0;i<n;i++)
    if(x.a[i]!=y.a[i])
     return x.a[i]<y.a[i];
   return 0;
  }
};
set <node,cmp> s;
node st;
node t;
int bfs(node f)
{
 queue <node> q;
 q.push(f);
 s.insert(f);
 int br=0;
 while(!q.empty())
  {
   node temp=q.front();
   q.pop();

   br++;
   int i;
  //  for(i=0;i<n;i++)
  //   printf("%d ",temp.a[i]);
   //  printf("\n");
  //   for(i=0;i<n;i++)
  //   printf("%d ",temp.b[i]);
   //  printf("\n");
   //  printf("\n");
   int l=0;
   for(i=0;i<n;i++)
    if(temp.a[i]!=st.a[i])
     { l=1;break;}
   if(l==0)
    for(i=0;i<n;i++)
     if(temp.b[i]!=st.b[i])
      {
       l=1;break;
      }
   if(l==0)return 1;
   node A;
   for(i=0;i<n;i++)
    {
     A.a[i]=temp.b[temp.a[i]-
1];
     A.b[i]=temp.b[i];
    }
   if(s.find(A)==s.end())
    {
     s.insert(A);
     q.push(A);
    }
   node B;
   for(i=0;i<n;i++)
    {
     B.b[i]=temp.a[temp.b[i]-
1];
     B.a[i]=temp.a[i];
    }
   if(s.find(B)==s.end())
    {
     s.insert(B);
     q.push(B);
    }
    if(br>10000)return 0;
  }
// printf("no\n");
 return 0;
}
int main()
{
 n=4;
```

```
st.a[0]=2;                t.a[2]=1;                  st.b[n-1]=n+1;
st.a[1]=1;                t.a[3]=4;                  st.b[n]=1;
st.a[2]=3;                                           t.a[n]=n+1;
st.a[3]=4;                t.b[0]=2;                  t.b[n-1]=n+1;
                          t.b[1]=3;                  t.b[n]=1;
st.b[0]=2;                t.b[2]=4;                  s.clear();
st.b[1]=3;                t.b[3]=1;                  }
st.b[2]=4;                for(n=4;n<=1000;n++)       return 0;
st.b[3]=1;                 {                         }
                           printf("%d          ==>
t.a[0]=2;                 %d\n",n,bfs(t));
t.a[1]=3;                  st.a[n]=n+1;
```

**Problem 10.** Are there any components, which are made of two patterns?

Solution.

If the component is made of two patterns, we have the following possibilities:

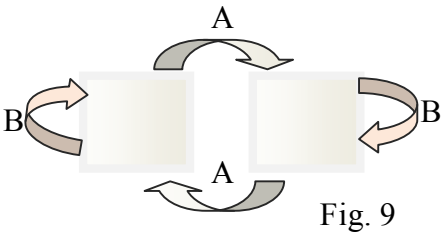



Fig. 9

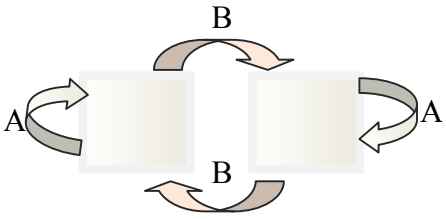



Fig. 10

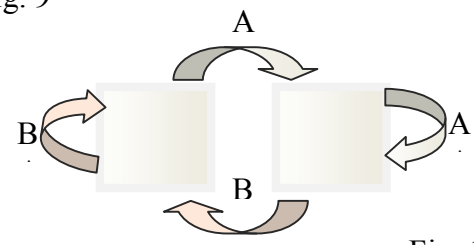



Fig. 11

Since we want using operations A or B to preserve the pattern, it must contain one line with the permutation 1 2 3..*n*.

We will prove that the third figure can not fulfils the conditions of a component of the graph. After using operation A on the first pattern, the second line is still the same and the first is changed. When we use operation A on the new pattern, the second line remains the same. Then, if it doesn't change into pattern 3, it should remains the same using operation A, or to transform in the previous one.

*Case 1.* If it remains the same, when we use operation B, the first line is changed and we get a new (third) pattern – contradiction.

*Case 2.* In this case, after using operation, after which the pattern is changing, the two rows become identical. But after an operation it should remain identical. Therefore, it also contains the permutation 1 2 3..*n*. Then the both patterns remain identical. If the next pattern transforms to the initial, than applying the operation B it must remain the same. Than we get two identical patterns, which construct component of one pattern. Therefore, it does not exist a component, containing of two patterns.

**Problem 11.** How many components, containing 3 patterns, are there in graph $G_n$ ?

Solution.

The pattern should contain in the both of its ends the permutation 1 2 3...*n*.
Then, it will be

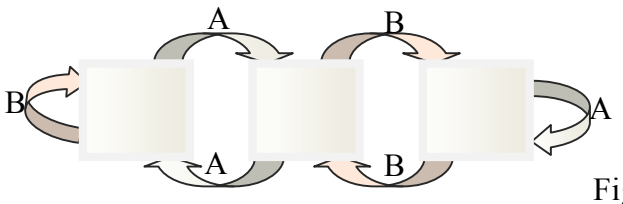



Fig. 12

In the other case, the component may have the following structure:

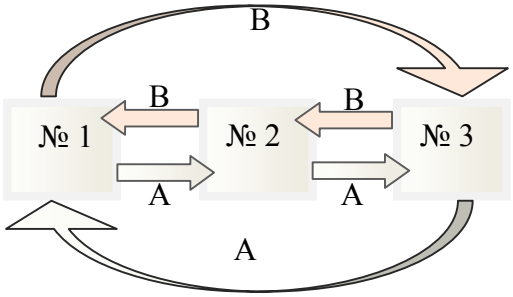



Fig. 13

In this case we are considering the pattern № 2. It turned out after applying operation B on the pattern № 3.Then, the first row doesn't change. After applying the operation B on the pattern №2 we get the pattern № 1 and the first row remain again the same. The three patterns are with identical first row. After we apply operation A on pattern №1, we get № 2, but in that case the second row remain the same. Likewise, we get № 3 from № 2. Therefore the only case is all the three patterns to be identical. But than we obtain a component, containing only one pattern.

*Deduction:* Only a component of the type shown on Fig. 12 can occur in the graph $G_n$.

We show that if a component contains 3 patterns, the both of its endmost patterns must have one row with the permutation 1 2 3 ... *n*-1 *n*. The number of the components, containing 3 patterns is *n*.

**Problem 12.** How many components contain the graph $G_n$?

Solution.

Each graph must contain a component, containing the pattern.

$$\begin{matrix} a_1\ a_2\ \dots\ a_n \\ b_1\ b_2\ \dots\ b_n \end{matrix}$$

The graph $G_3$ contains:

- 1 component with 1 pattern;
- 3 components with 3 patterns;

- 1 component with 8 patterns;
- 2 components with 9 patterns.

The number of all the components is 7.

**Conjecture.** The number of the components of the graph $G_n$ is $2^n - 1$.

# Second part

Denote by $H_n$ a graph, consisting of components that contain patterns with 3 rows and each row is a permutation of the numbers from 1 to $n$. On each pattern are allowed 6 operations:
A – Changing first row, in relation to the second one.
B – Changing first row, in relation to the third one.
C – Changing second row, in relation to the first one.
D – Changing second row, in relation to the third one.
E – Changing third row, in relation to the first one.
F – Changing third row, in relation to the second one.

**Problem 13.** Examine the graph $H_2$.
Solution.

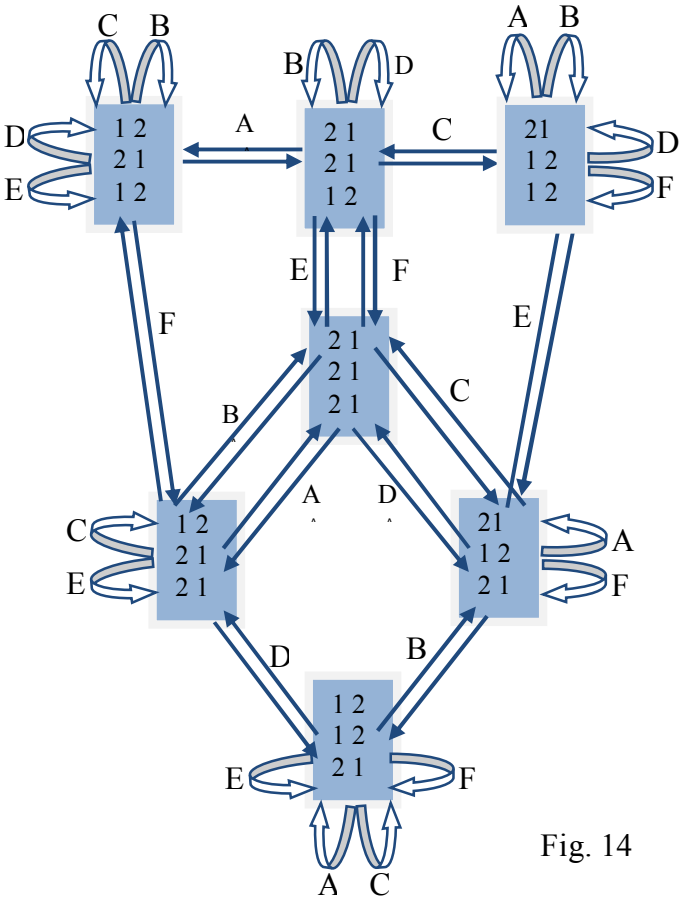



Fig. 14

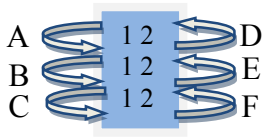


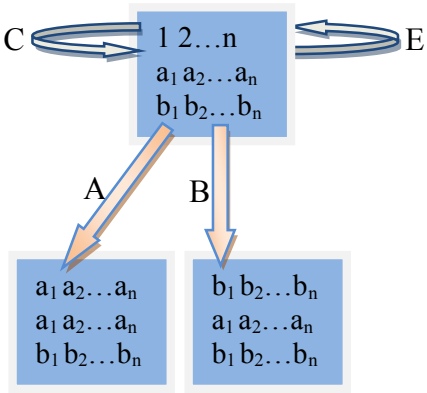

Fig. 15

Graph $H_2$ has all the permutations of the pattern, where $n = 2$, and they are $(n!)^3$. The graph has two components.

The first component contains 7 components, the second – only one component. This component contains the serial numbers 1 2 in each line and after using the operations A, B, C, D, E, F it doesn't change.
The two components are planar graphs with central symmetry.
From every pattern, which is vertex of the graph came out 6 edges and go in 6 edges again.

**Problem 14.** Investigate graph $H_3$.
Analysis.
The graph H3 has $(3!)^3 = 216$ patterns. It has one component with 1 pattern and other components, which are not investigated yet.

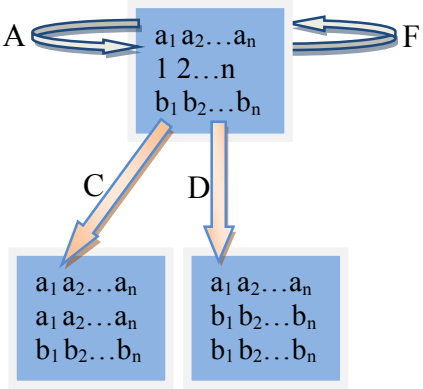
**Problem 15.** Investigate some patters, which contain the permutation of serial numbers *1 2… n-1 n* in the different lines.
Solution.
*Case 1.* If we have the permutation *1 2… n-1 n* in only one line.



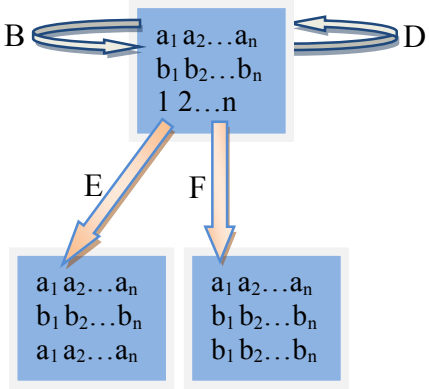

*a)* If the permutation of serial numbers *1 2… n-1 n* is on the first line, after using operations C and F the pattern doesn't chance and after using operations A and B, the permutation of the first line changes to the permutation of the second or third line (in dependence of the operation we use).
After using operation D and F the pattern changes to another different pattern.

It is analogy for the other cases when we have only one permutation *1 2… n-1 n* in the pattern.

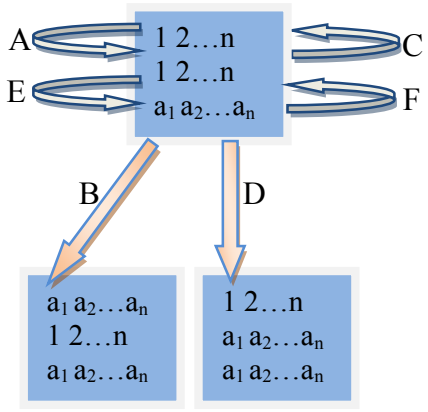
*b)*




*c)*



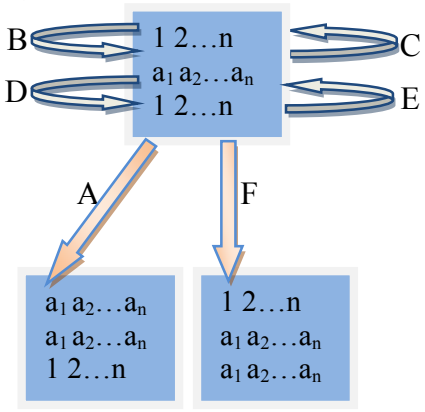
*Case 2.* If we have the permutation *1 2… n-1 n* in two lines of a pattern.



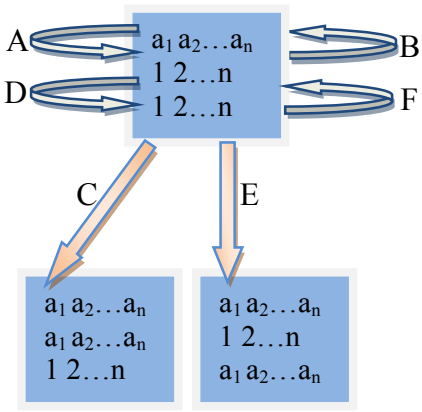

*a)* If the permutation of serial numbers *1 2… n-1 n* is on the first and second line, after using permutation A, C, E and F the pattern doesn't change, because we do operations with serial numbers. After using operations B and D, the pattern changes to other patterns, which have the permutation of serial numbers *1 2… n-1 n* in one of their lines. Then we return to *Case 1*, where we have investigated the cases.

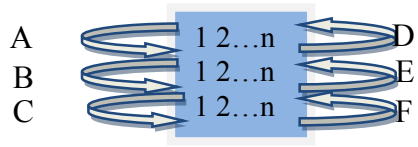
The other situations, where permutation of serial numbers *1 2… n-1 n* is located in two lines, are analogy.

*b)*




*c)*




*Case 3.* If the pattern has the permutation in the whole 3 lines, after using operations A, B, C, D, E, F the pattern doesn't change because the numbers in each line are serial and their place tally with the value of the number.

**Problem 16.** Find the number of patterns in a graph $H_n$.

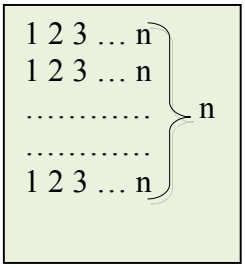
Solution.

In a pattern of the graph $H_n$ we have 3 lines and n digits in each line. So the number of all patterns in graph $H_n$ is the number of permutations of the patterns. It is the number of permutations of each line – n!, raised on third power $(n!)^3$.

**Problem 17.** Find the number of patterns in a graph $N_n$ (a graph which has n lines and n digits in each line).

Solution.
When we have n digits in each line, the number of permutations of each line is n!. Consequently, if we have n lines, the number of permutations of the pattern is $(n!)^n$.

**Problem 18.** How does the component in the graph $H_n$ changes after using the operations, if we have the permutation of serial numbers *1 2… n-1 n* in each line of the pattern?




Solution.

The when we use a random operation on a random line, it doesn't change, because they are serial and after replacing the numbers their order stays the same.

**SUMMARY**

In the development we have achieved the following results:
The proposed solutions and the given examples are fully copyright lawsuit.

# Part one

1. Founded numbers of vertexes in graph $G_3$, generalized for graph $G_n$.
2. Examined components and their geometrical properties in graph $G_3$ and generalized for $G_n$.
3. Proved a problem about the connection between patterns, edges, and areas in a planar graph.
4. Examined some properties of specific patterns.
5. Given an evidence for obtaining patterns.
6. Included a program about obtaining the patterns.
7. Examined elements with two and three patterns.
8. Given a hypothesis about the number of components in graph $G_n$.

# Part two

1. Examined a graph $H_2$.
2. Examined a graph $H_3$.
3. Examined a graph $H_n$.
4. Examined a graph $N_n$.

**Literature.**

[1] Математическа библиотека, СМБ, Подготовка за олимпиади, София, 2002.
[2] Б. Болобаш, Теория на графите, Наука и изкуство, София, 1989.