УДК 519.17+004.021

ЗАДАЧА О СУЩЕСТВОВАНИИ ДЕРЕВА С ХАРАКТЕРИСТИЧЕСКИМ ВЕКТОРОМ УЗЛОВЫХ ВЕРШИН

И. Н. Попов[1]

САФУ имени М.В. Ломоносова, г. Архангельск

[1]PopovIvanNik@yandex.ru

Аннотация

В работе рассматривается задача о существовании дерева с определенными числовыми характеристиками. Если задано дерево, то можно определить количества узловых вершин дерева и листьев, а также определить их степени. Тем самым для дерева можно определить набор пар, составные которых есть числа, соответствующие количествам узловых вершин и их степеней. Можно сформулировать обратную задачу: задаются пары натуральных чисел, вторые компоненты которых больше 1, и следует определить, найдется ли хотя бы одно дерево, что количества его узловых вершин и их степеней совпадают с данными парами. Решение этой задачи представлено в данной работе.

Ключевые слова*: алгоритм, Python, граф-дерево, код Прюфера дерева*

1. КОД ПРЮФЕРА ДЕРЕВА

Под графом понимается псевдограф, не содержащий кратных ребер и петель [3]. Связный граф, не содержащий циклов, называется деревом; вершины степени 1 в дереве называются листьями; остальные вершины называются узловыми. Будем рассматривать помеченные деревья – деревья, в которых каждой вершине сопоставлено натуральное число от 1 до количества вершин в дереве.

Обозначим дерево $G = (V; X)$, где $V = U \cup L$ – множество узловых вершин U и множество листьев L, X – множество ребер.

Один из способов хранения дерева, как и всех псевдографов, является запись информации о дереве в матрицу смежности. В матрице смежности A дерева с n вершинами количество единиц равно $2(n-1)$. Действительно, так как в дереве число вершин на 1 больше ребер, то в дереве с n вершинами содержится $n-1$ ребро; и так как матрица смежности симметрична относительно главной диагонали, на которой расположены нули, в матрице A дерева $2(n-1)$ единиц.

Получаем, что «концентрация» единиц в матрице смежности A дерева с n вершинами (то есть отношение числа единиц к числу элементов в матрице) не превосходит величины $\frac{2}{n+1}$, так как

$$\frac{2(n-1)}{n^2} < \frac{2(n-1)}{n^2-1} = \frac{2}{n+1}.$$

Например, при $n = 7$ получаем, что число единиц в матрице A меньше 25% от общего числа элементов матрицы; чем больше n, тем меньше этот процент.

Имеются другие способы хранения деревьев, среди которых выделим хранение дерева в коде Прюфера.

Алгоритм кодирования дерева. Исходные данные: дерево с n вершинами. Выходные данные: набор из $n-2$ чисел.

1. Выбрав в дереве лист с наименьшим номером, он удаляется вместе с инцидентным с ним ребром и выписывается номер вершины, стоящей у другого конца удаленного ребра. Это число и есть первый символ кода Прюфера данного дерева. 2. Повторяются аналогичные операции над остатками дерева до тех пор, пока не получится граф с двумя вершинами. Полученный набор знаков и есть код Прюфера, соответствующий исходному дереву.

Длина кода Прюфера дерева $G = (V; X)$ равна $|V| - 2$ и код состоит из чисел, принадлежащих множеству $\{1; 2; \dots; |V|\}$. Число кодов Прюфера совпадает с числом всех помеченных деревьев и равно $|V|^{|V|-2}$ (теорема Кэли).

Пример. Для дерева, изображенного на рисунке 1, код Прюфера 7542714.

$$\begin{array}{ccccccccc} 3 & - & 7 & - & 1 & - & 4 & - & 9 \\ & & | & & & & | & & \\ 8 & - & 2 & & & & 5 & - & 6 \end{array}$$

Рис. 1. Дерево с кодом 7542714

В коде Прюфера номер вершины дерева встречается на 1 меньше степени самой вершины (поэтому, учитывая, что у листьев степени равны 1, в записи кода Прюфера номера листьев не используются). ■

Рассмотрим способ нахождения кода Прюфера дерева $G = (V; X)$, заданного набором своих ребер.

По элементам множества X строится таблица Δ степеней вершин дерева, в первой строке которой записаны вершины $v_1, v_2, \ldots, v_n$ по возрастанию их номеров, во второй строке – степени вершин $\delta_1, \delta_2, \ldots, \delta_n$. По первому слева элементу δ_i таблицы Δ, равному 1, определяется лист v_i наименьшего номера. Во множестве X однозначно определяется ребро $\{v_i; v_j\}$. Тогда j – первый номер кода Прюфера дерева G. Ребро $\{v_i; v_j\}$ удаляется из множества X и числа δ_i и δ_j таблицы Δ уменьшаются на 1. Затем процесс повторяется и до тех пор, пока в таблице Δ не получится только две единицы (остальные элементы будут равны 0).

Например, пусть $X = \{\{1;3\};\{3;2\};\{3;6\};\{4;2\};\{5;6\}\}$. Тогда получаем следующие шаги нахождения кода Прюфера дерева с данным множеством ребер:

- $\Delta = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ \underline{1} & 2 & \underline{3} & 1 & 1 & 2 \end{pmatrix}, \{\cancel{\{1;3\}};\{3;2\};\{3;6\};\{4;2\};\{5;6\}\};$
- $\Delta = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 0 & \underline{2} & 2 & \underline{1} & 1 & 2 \end{pmatrix}, \{\{3;2\};\{3;6\};\cancel{\{4;2\}};\{5;6\}\};$
- $\Delta = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 0 & \underline{1} & \underline{2} & 0 & 1 & 2 \end{pmatrix}, \{\cancel{\{3;2\}};\{3;6\};\{5;6\}\};$
- $\Delta = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 0 & 0 & \underline{1} & 0 & 1 & \underline{2} \end{pmatrix}, \{\cancel{\{3;6\}};\{5;6\}\};$
- $\Delta = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}, \{\{5;6\}\}.$

Код Прюфера составляют номера вершин, определяемые по подчеркнутым числам, отличных от 1, таблиц Δ, получаемых на каждом этапе: 3236.

На Python [5] генерация кода Прюфера по набору ребер дерева осуществляется с использованием указанного способа:

```
def codPrufera(v):
    codPv=[]
    nm=len(set(v)); vs=[]
    for i in range(nm): vs.append(0)
    n=len(v)
    for i in range(n): vs[v[i]-1]=vs[v[i]-1]+1
    for k in range(nm-2):
        nvs=vs.index(1); nn=v.index(nvs+1)
        if v[2*int(nn/2)+1] == v[nn]: pl=v[2*int(nn/2)]
        else: pl=v[2*int(nn/2)+1]
```

```
        v[2*int(nn/2)+1]=0; v[2*int(nn/2)]=0
        vs[nvs]=vs[nvs]-1; vs[pl-1]=vs[pl-1]-1; codPv.append(pl)
    return(codPv)
```

Например, набор команд

```
vr=[7,2,10,4,5,8,4,11,5,4,1,5,6,1,6,12,7,6,3,7,2,9,3,13]
print(codPrufera(vr))
```

приводит к получению кода Прюфера [5,2,7,4,4,5,1,6,6,7,3] дерева, ребра которого последовательно собраны в вектор vr.

Алгоритм декодирования дерева. Исходные данные: набор из $n-2$ чисел. Выходные данные: дерево с n вершинами.

1. На плоскости изображаются n пронумерованных точек. 2. Выписывается под кодом в порядке возрастания все те из чисел $1, 2, \dots, n$, которые не входят в код. Полученный набор чисел называется антикодом. 3. Точка, отмеченная первым числом кода, соединяется с точкой, отмеченной первым числом антикода, после чего вычеркиваются эти числа из кода и антикода. 4. Если вычеркнутое число кода не встречается в оставшейся части кода, его вписывают на соответствующее место в антикоде (числа в антикоде упорядочены по возрастанию). 5. Аналогичные операции повторяются с новыми кодами и антикодами до тех пор, пока все числа в коде не окажутся вычеркнутыми. 6. Наконец соединяются две точки, отмеченными номерами, совпадающими с числами последнего антикода. Полученное дерево и есть искомое.

Пример. Полный граф – это граф, в котором все вершины друг с другом соединены ребрами. Количество ребер в полном графе с n вершинами равно $\frac{n(n-1)}{2}$. У полного графа с n вершинами существует n^{n-2} деревьев, каждое из которых является его остовом; каждый остов содержит $n-1$ ребро (отсюда сумма степеней всех вершин остова равна $2(n-1)$; по крайней мере две степени равны 1; наибольшая степень равна $n-1$ для пучка ребер – дерева, все ребра которого инцидентны одной вершине).

Каждому остову полного графа сопоставляется код Прюфера, как последовательность натуральных чисел от 1 до n, состоящая из $n-2$ чисел с возможными их повторениями. Последовательности различаются как порядком следования в ней чисел, так и составом. Поэтому коды Прюфера – это размещения с повторениями [1]. Сгенерировав все размещения с повторениями длины $n-2$ из чисел множества $\{1; 2; \ldots; n\}$, получаем все коды Прюфера остовов полного графа; декодируя коды, получаем сами остовы (например, в виде наборов их ребер).

На языке Python

● генерация множества ребер дерева по коду Прюфера:

```
def MRD(CP):
    n=len(CP)+2
    ACP=[]; ACP=list(set(range(1,n+1)) - set(CP))
    P=[]
    while len(CP) != 0:
        P.append([CP[0],ACP[0]])
        ACP.pop(0)
        CPV=[]; CPV.extend(CP); CPV.pop(0);
        if {CP[0]} & set(CPV) == set(): ACP.append(CP[0]); ACP.sort()
        CP.pop(0)
    P.append(ACP)
    return(P)
```

● генерации размещений с повторениями:

```
def GRP(k,n):
    b=[];L=[]
    for i in range(k): b.append(1)
    while b[k-1] != n+1:
        L.extend(b)
        i=0
        while i<k-1 and b[i]==n: b[i]=1; i=i+1
        b[i]=b[i]+1
    return(L)
```

● генерация кода Прюфера из списка с данным номером:

```
def NLCP(k,n,NCP):
    GRP(k,n)
    LCP=[]
```

```
    for i in range(k): LCP.append(L[(NCP-1)*k+i])
    return(LCP)
```

Например, выполнение команды print(MRD([4,6,4,2])) дает результат

[[4, 1], [6, 3], [4, 5], [2, 4], [2, 6]];

результат – это набор ребер дерева с кодом Прюфера [4,6,4,2] (рис. 2).

$$\begin{array}{ccccccccc}3&-&6&-&2&-&4&-&5\\&&&&&&|&&\\&&&&&&1&&\end{array}$$

Рис. 2. Дерево с кодом Прюфера [4,6,4,2]

Выполнение команды for i in range(16): print(MRD(NLCP(2,4,i+1))) приводит к получению наборов ребер остовов полного графа с 4-мя вершинами:

[[1, 2], [1, 3], [1, 4]]	[[2, 3], [1, 2], [1, 4]]	[[3, 2], [1, 3], [1, 4]]	[[4, 2], [1, 3], [1, 4]]
[[1, 3], [2, 1], [2, 4]]	[[2, 1], [2, 3], [2, 4]]	[[3, 1], [2, 3], [2, 4]]	[[4, 1], [2, 3], [2, 4]]
[[1, 2], [3, 1], [3, 4]]	[[2, 1], [3, 2], [3, 4]]	[[3, 1], [3, 2], [3, 4]]	[[4, 1], [3, 2], [3, 4]]
[[1, 2], [4, 1], [3, 4]]	[[2, 1], [4, 2], [3, 4]]	[[3, 1], [4, 2], [3, 4]]	[[4, 1], [4, 2], [3, 4]]

Остовы в полном графе с n вершинами могут рассматриваться и как базисы симметрической группы S_n – группы подстановок длины n с композицией в качестве бинарной алгебраической операции [2, 4]. Базис группы S_n – независимая система образующих, все элементы которой являются транспозициями, число которых равно $\frac{n(n-1)}{2}$, что совпадает с числом ребер в полном графе с n вершинами.

Множеству Ω из транспозиций симметрической группы S_n сопоставляется граф Пойа $G(\Omega) = (V; X)$, в котором $V = \{1; 2; \dots; n\}$ – множество вершин и ребро $\{i; j\} \in X$, если транспозиция $(ij) \in \Omega$, где i и j – попарно различные числа из множества V. Справедливо: множество Ω, состоящее из транспозиций группы S_n, является системой образующих группы S_n тогда и только тогда, когда граф Пойа $G(\Omega)$ является связным; множество Ω транспозиций группы S_n является базисом группы S_n тогда и только тогда, когда граф Пойа $G(\Omega)$ является деревом. Отсюда следует, что в любом базисе группы S_n содержится $n - 1$ транспозиция и количество базисов равно n^{n-2}, что равно числу помеченных деревьев с n вершинами.

Например, множество $\Omega = \{(14); (16); (17); (18); (28); (38); (58)\}$ – базис группы S_8; множеству Ω соответствует дерево Пойа $G(\Omega)$ (рис. 3).

$$\begin{array}{ccccccc} & & 4 & & 2 & & \\ & & | & & | & & \\ 7 & - & 1 & - & 8 & - & 5 \\ & & | & & | & & \\ & & 6 & & 3 & & \end{array}$$

Рис. 3. Граф Пойа $G(\Omega)$

Каждую подстановку из группы S_8 можно представить в виде композиции транспозиций из множества Ω. Например,

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 6 & 2 & 1 & 8 & 7 & 3 & 4 & 5 \end{pmatrix} = (16)(18)(83)(18)(41)(85)(18)(17)(14).$$

Генерируя все коды Прюфера и составляя по каждому из них множество ребер дерева, получаем все базисы симметрической группы S_n. ■

2. НЕКОТОРЫЕ ЧИСЛОВЫЕ ХАРАКТЕРИСТИКИ ДЕРЕВА

Будем считать, что в дереве содержится m_1 вершин степени k_1, m_2 вершин степени k_2 и так далее, и m_s вершин степени k_s, где $k_1, k_2, \dots, k_s$ – попарно различные натуральные числа большие 1, и других вершин, кроме листьев, нет. Тогда количество листьев $|L|$ в дереве равно

$$|L| = m_1(k_1 - 2) + m_2(k_2 - 2) + \dots + m_s(k_s - 2) + 2.$$

Действительно, так как сумма степеней всех вершин в произвольном псевдографе равна удвоенному количеству ребер и в дереве число ребер на 1 меньше числа вершин, то справедливо равенство

$$m_1k_1 + m_2k_2 + \dots + m_sk_s + |L| = 2 \cdot \big((m_1 + m_2 + \dots + m_s + |L|) - 1\big),$$

из которого и получается требуемое.

Так как число вершин в дереве равно $m_1 + m_2 + \dots + m_s + |L|$, то количество вершин $|V|$ в дереве равно

$$|V| = m_1(k_1 - 1) + m_2(k_2 - 1) + \dots + m_s(k_s - 1) + 2.$$

Видим, что число листьев и общее количество вершин в дереве выражаются через количества узловых вершин дерева. Вектор $(m_1; k_1; m_2; k_2; \dots; m_s; k_s)$ назовем характеристическим вектором узловых вершин дерева. Формально можно рассматривать вектор, который не содержит ни одного элемента; этому вектору соответствует дерево с двумя вершинами и одним ребром.

3. ОСНОВНАЯ ЗАДАЧА

Пусть $(m_1; k_1), (m_2; k_2), \dots, (m_s; k_s)$ – пары натуральных чисел и $k_1, k_2, \dots, k_s$ попарно различны и больше 1. Сформулируем задачу о существовании дерева с характеристическим вектором $(m_1; k_1; m_2; k_2; \dots; m_s; k_s)$ его узловых вершин.

По вектору $(m_1; k_1; m_2; k_2; \dots; m_s; k_s)$ составим код Прюфера $c = c_1 c_2 \dots c_s$,

$$c_1 = \underbrace{1, \dots, 1}_{k_1-1}, \underbrace{2, \dots, 2}_{k_1-1}, \dots, \underbrace{m_1, \dots, m_1}_{k_1-1};$$

$$c_2 = \underbrace{m_1+1, \dots, m_1+1}_{k_2-1}, \underbrace{m_1+2, \dots, m_1+2}_{k_2-1}, \dots, \underbrace{m_1+m_2, \dots, m_1+m_2}_{k_2-1};$$

$$c_3 = \underbrace{m_1+m_2+1, \dots, m_1+m_2+1}_{k_3-1}, \underbrace{m_1+m_2+2, \dots, m_1+m_2+2}_{k_3-1}, \dots,$$
$$\underbrace{m_1+m_2+m_3, \dots, m_1+m_2+m_3}_{k_3-1}$$

и так далее. В общем случае для каждого $i \in \{1; 2; \dots; s\}$ справедливо:

$$c_i = \underbrace{d_i+1, \dots, d_i+1}_{k_i-1}, \underbrace{d_i+2, \dots, d_i+2}_{k_i-1}, \dots, \underbrace{d_i+m_i, \dots, d_i+m_i}_{k_i-1},$$

$$d_1 = 0, \qquad d_i = \sum_{j=1}^{i-1} m_j, \text{если } i \geq 2.$$

Каждое c_i состоит из m_i блоков с $k_i - 1$ элементом, поэтому длина кода равна $m_1(k_1-1) + m_2(k_2-1) + \dots + m_s(k_s-1)$, то есть совпадает с числом $|V| - 2$. Наибольшее число в коде c равно $m_1 + m_2 + \dots + m_s$, которое меньше числа $|V|$.

Пример. Из пар $(3; 5), (2; 7), (2; 6)$ образуем вектор $(\overline{3}; \underline{5}; \overline{2}; \underline{7}; \overline{2}; \underline{6})$. По вектору составим последовательность (с согласованием нижнего и верхнего подчеркивания элементов в векторе и числа блоков и элементов в последовательности):

$$c = \overbrace{\underbrace{1,1,1,1}_{\underline{5}-1=4}, \underbrace{2,2,2,2}_{\underline{5}-1=4}, \underbrace{3,3,3,3}_{\underline{5}-1=4}}^{\overline{3}}, \overbrace{\underbrace{4,4,4,4,4,4}_{\underline{7}-1=6}, \underbrace{5,5,5,5,5,5}_{\underline{7}-1=6}}^{\overline{2}}, \overbrace{\underbrace{6,6,6,6,6}_{\underline{6}-1=5}, \underbrace{7,7,7,7,7}_{\underline{6}-1=5}}^{\overline{2}}.$$

Данную последовательность будем рассматривать как код Прюфера дерева, в котором содержится 36 вершин (длина кода равна 34). ■

На языке Python генерации кода Прюфера по характеристическому вектору узловых вершин дерева осуществляется следующим образом:

```
def GCP(W):
    C=[]; d=1
    for i in range(int(len(W)/2)):
        for k in range(W[2*i]):
            for p in range(W[2*i+1]-1): C.append(k+d)
        d=d+W[2*i]
    return(C) #Код Прюфера дерева
```

Запуск команды print(GCP([3,2,4,3])) приводит к результату

[1, 2, 3, 4, 4, 5, 5, 6, 6, 7, 7];

интерпретация результата: по характеристическому вектору [3,2,4,3] дерева (в котором 3 вершины со степенями 2 и 4 вершины со степенями 3) генерируется код Прюфера [1,2,3,4,4,5,5,6,6,7,7]. Команда print(MRD(GCP([3,2,4,3]))) позволяет получить следующий результат:

[[1, 8], [2, 1], [3, 2], [4, 3], [4, 9], [5, 4], [5, 10], [6, 5], [6, 11], [7, 6], [7, 12], [7, 13]];

интерпретация результата: вначале по характеристическому вектору [3,2,4,3] вершин дерева генерируется код Прюфера дерева [1, 2, 3, 4, 4, 5, 5, 6, 6, 7, 7], затем по этому коду получается набор [[1, 8], [2, 1], [3, 2], [4, 3], [4, 9], [5, 4], [5, 10], [6, 5], [6, 11], [7, 6], [7, 12], [7, 13]] ребер дерева (рис. 4).

$$
\begin{array}{ccccccccccccccccc}
8 & - & 1 & - & 2 & - & 3 & - & 4 & - & 5 & - & 6 & - & 7 & - & 13 \\
 & & & & & & & & | & & | & & | & & | & & \\
 & & & & & & & & 9 & & 10 & & 11 & & 12 & &
\end{array}
$$

Рис. 4. Дерево с вектором [3,2,4,3] числа узловых вершин и их степеней

ЗАКЛЮЧЕНИЕ

В работе показано применение кода Прюфера для решения задач, связанных с деревьями: генерация остовов полного графа; генерация базисов симметрической группы; доказательство существования дерева с заданным характеристическим вектором узловых вершин. Приведены программы на языке Python для определения набора ребер дерева по его коду Прюфера и порождения кода Прюфера по характеристическому вектору узловых вершин.

СПИСОК ЛИТЕРАТУРЫ

1. *Иванов Б.Н.* Дискретная математика. Алгоритмы и программы. М.: Лаборатория базовых знаний, 2001. 288 с.
2. *Калужнин Л.А., Сущанский В.И.* Преобразования и перестановки. М.: Наука. Главная редакция физико-математической литературы, 1985. 160 с.
3. *Нефедов В.Н., Осипова В.А.* Курс дискретной математики. М.: Изд-во МАИ, 1992. 264 с.
4. *Попов И.Н, Попова А.В.* Разложение подстановок в циклы: реализация приложений с помощью Python // Сборник трудов Международной научной конференции «Актуальные проблемы прикладной математики, информатики и механики», Воронеж, Россия, 11-13 ноября 2019 г. Воронеж: Издательство «Научно-исследовательские публикации», 2020. С. 484-491.
5. *Lutz M.* Learning Python, 4th Edition. O'Reilly Media, Inc. 2011. 1280 p.

THE PROBLEM OF THE EXISTENCE OF A TREE WITH A CHARACTERISTIC VECTOR OF NODE VERTICES

I. N. Popov[1]

Department of Higher Mathematics Northern (Arctic) Federal University named after M.V. Lomonosov, Arkhangelsk

[1]PopovIvanNik@yandex.ru

Abstract

The paper presents the problem of the existence of a tree with certain numerical characteristics. It is clear that if a tree is given, it is possible to determine the number of node vertices of the tree and leaves, as well as to determine their degrees. Thus, for a tree, you can define a set of pairs whose coordinates are numbers corresponding to the number of node vertices and their degrees. We can form the inverse problem: we give pairs of natural numbers whose second coordinates are greater than 1, and we should determine whether there is at least one tree that the numbers of its node vertices and their degrees coincide with these pairs. The solution to this problem is presented in this paper.

***Keywords**: algorithm, Python, graph-tree, Prufer code of the tree*

REFERENCES

1. *Ivanov B.N.* Diskretnaya matematika. Algoritmi i programmi. M.: Laboratoriya bazovih znanii, 2001. 288 s.
2. *Kalujnin L.A., Suschanskii V.I.* Preobrazovaniya i perestanovki. M.: Nauka. Glavnaya redakciya fiziko-matematicheskoi literature, 1985. 160 s.
3. *Nefedov V.N., Osipova V.A.* Kurs diskretnoi matematiki. M.: Izd-vo MAI, 1992. 264 s.
4. *Lutz M.* Learning Python, 4th Edition. O'Reilly Media, Inc. 2011. 1280 p.
5. *Popov I.N, Popova A.V.* Razlojenie podstanovok v cikli: realizaciya prilojenii s pomoschyu Python // Sbornik trudov Mejdunarodnoi nauchnoi konferencii «Aktualnie problemi prikladnoi matematiki, informatiki i mehaniki», Voronej, Rossiya, 11-13 noyabrya 2019 g. Voronej: Izdatelstvo «Nauchno-issledovatelskie publikacii», 2020. S. 484-491.

СВЕДЕНИЯ ОБ АВТОРАХ

ПОПОВ Иван Николаевич – к.ф.-м.н., доцент, доцент кафедры Высшей математики, САФУ имени М.В. Ломоносова, г. Архангельск

Ivan Nikolaevich Popov – PhD in Physico-mathematical sciences, Assoc. Prof., PhD, Department of Higher Mathematics, Northern (Arctic) Federal University named after M.V. Lomonosov, Arkhangelsk

email: PopovIvanNik@yandex.ru

Материал поступил в редакцию 04 февраля 2021 года