

Об алгоритмических методах исследования двухцветных раскрасок гиперграфов

А. В. ЛЕБЕДЕВА

Московский государственный университет
им. М. В. Ломоносова
e-mail: anuta278@yandex.ru

УДК 519.179.1+519.174.7

Ключевые слова: гиперграф, раскраска, хроматическое число.

Аннотация

Рассматривается экстремальная задача о раскрасках гиперграфов. Пусть k — натуральное число. Требуется найти величину $m_k(n)$, равную минимальному количеству рёбер n -однородного гиперграфа, не допускающего таких двухцветных раскрасок множества вершин, что в каждом ребре гиперграфа содержатся по крайней мере k вершин каждого цвета. В работе получены верхние оценки величин $m_k(n)$ для малых значений k , n , найдено значение $m_4(8)$, получена нижняя оценка $m_3(7)$.

Abstract

A. V. Lebedeva, On algorithmic methods of analysis of two-colorings of hypergraphs, Fundamentalnaya i prikladnaya matematika, vol. 19 (2014), no. 2, pp. 125–149.

This paper deals with an extremal problem concerning hypergraph colorings. Let k be an integer. The problem is to find the value $m_k(n)$ equal to the minimum number of edges in an n -uniform hypergraph not admitting two-colorings of the vertex set such that every edge of the hypergraph contains at least k vertices of each color. In this paper, we obtain upper bounds of $m_k(n)$ for small k and n , the exact value of $m_4(8)$, and a lower bound for $m_3(7)$.

1. Введение и формулировки результатов

В настоящей работе рассматривается обобщение известной экстремальной задачи о раскрасках однородных гиперграфов.

Сперва напомним определения основных объектов и постановку классической задачи. *Гиперграфом* называется пара множеств $H = (V, E)$, где V — некоторое конечное множество, называемое *множеством вершин* гиперграфа, а E — совокупность подмножеств множества V , которые называются *рёбрами* гиперграфа. Мы предполагаем, что в гиперграфе нет вершин степени 0, т. е. вершин, не принадлежащих ни одному ребру. Гиперграф называется *n -однородным*, если каждое его ребро содержит ровно n вершин. Одной из классических

экстремальных задач теории гиперграфов является задача о свойстве В гиперграфов. Гиперграф обладает *свойством В*, если существует двухцветная раскраска множества его вершин, в которой все рёбра гиперграфа являются неодноразноцветными. Классическая задача, предложенная П. Эрдёшем и А. Хайналем в 1961 году [13], состоит в отыскании величины $m(n)$, равной минимальному числу рёбер гиперграфа в классе n -однородных гиперграфов, не обладающих свойством В.

Для задачи Эрдёша—Хайналя было получено множество различных результатов и предложен ряд важных обобщений [1]. В настоящей работе мы занимаемся обобщением для задачи Эрдёша—Хайналя, связанным со свойством B_k гиперграфов, которое было впервые определено в [7]. Гиперграф обладает *свойством B_k* , если существует двухцветная раскраска множества его вершин, в которой любое ребро гиперграфа содержит не менее k вершин каждого цвета. Требуется найти величину $m_k(n)$, равную минимальному числу рёбер гиперграфа в классе n -однородных гиперграфов, не обладающих свойством B_k .

В [2, 5, 8, 9] были найдены следующие асимптотические оценки величины $m_k(n)$ при различных условиях на параметр k : если $k = o(n)$,

$$m_k(n) \leq (1 + o(1)) \frac{e \ln 2}{4} n^2 \frac{2^n}{C_n^{k-1}};$$

при $k = O(\ln n)$ выполняется соотношение

$$m_k(n) = \Omega \left(\left(\frac{n}{\ln n} \right)^{1/2} \frac{e^{-k/2}}{\sqrt{2k-1}} \frac{2^{n-k}}{C_n^{k-1}} \right);$$

если k и n удовлетворяют неравенству $k^2(n-k) < (n-2k)^2$, то

$$m_k(n) \geq 0,19 n^{1/4} \frac{2^n}{C_n^{k-1}}.$$

Однако эти результаты не дают хороших оценок при малых значениях параметров. Наша работа посвящена оценкам величин $m_k(n)$ при некоторых малых k и n . Подобные результаты были получены ранее в [3, 4, 10–12]. В частности, в [3] было доказано, что $m_2(5) = 7$, $m_2(4) = 4$.

В настоящей работе получены верхние оценки величин $m_k(n)$ для малых значений k , n и получена нижняя оценка $m_3(7)$.

Теорема 1. *Имеет место оценка $m_3(7) \geq 6$.*

В [6] исследовалась величина $m_{n/2}(n)$ при растущем n . В настоящей работе найдено значение $m_4(8)$.

Теорема 2. *Имеет место равенство $m_4(8) = 4$.*

Доказательство теоремы 1 будет проведено в разделе 2, доказательство теоремы 2 — в разделе 3. Кроме того, мы разработали алгоритмический подход, который позволил получить ряд верхних оценок величины $m_k(n)$. В [4] были представлены следующие верхние оценки.

k/n	2	3	4	5	6	7	8	9	10
1	3	7	23	51	147	421	1 339	2 401	7 803
2	—	—	4	7	36	106	380	1 104	2 401
3	—	—	—	—	3	8	48	147	147

При помощи нашего алгоритмического подхода нам удалось улучшить эти результаты.

Теорема 3. Имеют место оценки $m_2(6) \leq 22$, $m_2(7) \leq 69$, $m_2(8) \leq 318$, $m_3(8) \leq 16$, $m_3(9) \leq 90$.

Теорема 3 приводит к новой таблице верхних оценок.

k/n	2	3	4	5	6	7	8	9	10
1	3	7	23	51	147	421	1 339	2 401	7 803
2	—	—	4	7	22	69	318	1 104	2 401
3	—	—	—	—	3	8	16	90	147

2. Доказательство теоремы 1

2.1. Схема доказательства

Чтобы доказать теорему, нам нужно установить, что при $|E| = 5$ гиперграф H обладает свойством B_3 . Итак, будем рассматривать 7-однородные гиперграфы с $|E| = 5$.

Введём ещё несколько дополнительных определений. *Половинной раскраской* множества вершин V будем называть его двухцветную раскраску, в которой ровно $\lfloor |V|/2 \rfloor$ вершин покрашены в один цвет, а остальные $\lceil |V|/2 \rceil$ — в другой. Назовём ребро n -однородного гиперграфа $H = (V, E)$ *k -хорошим* в двухцветной раскраске χ , если в этой раскраске оно содержит по крайней мере k вершин одного цвета и k другого. Иначе будем говорить, что ребро *k -плохое*. Из определения следует, что гиперграф обладает свойством B_k , если существует двухцветная раскраска множества его вершин, в которой все рёбра являются k -хорошими.

Для доказательства теоремы 1 нам достаточно обосновать следующие леммы.

Лемма 1. Пусть 7-однородный гиперграф $H = (V, E)$ таков, что $|V| \leq 10$, $|E| = 5$. Тогда H обладает свойством B_3 .

Лемма 2. Пусть 7-однородный гиперграф $H = (V, E)$ таков, что $|V| = 11$, $|E| = 5$. Тогда H обладает свойством B_3 .

Лемма 3. Пусть 7-однородный гиперграф $H = (V, E)$ таков, что $|V| \geq 12$, $|E| = 5$. Тогда H обладает свойством B_3 .

Доказательство леммы 1 проведём в разделе 2.2, доказательство леммы 2 — в разделе 2.3, доказательство леммы 3 — в разделе 2.4.

2.2. Доказательство леммы 1

Пусть $H = (V, E)$ — 7-однородный гиперграф, $|V| \leq 10$, $|E| = 5$. Докажем, что H обладает свойством B_3 . Для этого рассмотрим два случая в зависимости от количества вершин, но предварительно сделаем несложное замечание.

Замечание 1. Пусть W — некоторая совокупность двухцветных раскрасок множества V , $W(e)$ — множество раскрасок из W , в которых ребро e является 3-плохим. Пусть выполнено соотношение

$$\left| \bigcup_{e \in E} W(e) \right| < |W|.$$

Тогда H обладает свойством B_3 .

Доказательство очевидно.

СЛУЧАЙ 1. Пусть $|V| = 9$ (или $|V| = 10$). Существует 252 половинных раскраски 9-элементного (10-элементного) множества. При этом для каждого ребра существует ровно 42 половинных раскраски, в которых оно является 3-плохим. Тогда имеем $5 \cdot 42 = 210 < 252 = |W|$. Следовательно, по замечанию 1 гиперграф H обладает свойством B_3 .

СЛУЧАЙ 2. Если $|V| = 8$ (или $|V| = 7$), то при любой половинной раскраске ни одно ребро не может содержать более четырёх вершин одного цвета, т. е. все рёбра H являются 3-хорошими и H обладает свойством B_3 . Лемма доказана.

2.3. Доказательство леммы 2

Пусть $H = (V, E)$ — гиперграф из условия леммы 2, т. е. H — 7-однородный гиперграф, удовлетворяющий условиям $|V| = 11$, $|E| = 5$. Мы разобьём доказательство леммы 2 на несколько частей.

2.3.1. Схема доказательства

Утверждение 1. Пусть W — множество половинных раскрасок гиперграфа $H = (V, E)$, $e, f \in E$. Тогда $|W| = 924$ и

- 1) $|W(e)| = 224$;
- 2) если $|e \cap f| = 3$, то $|W(e) \cap W(f)| = 60$;
- 3) если $|e \cap f| = 4$, то $|W(e) \cap W(f)| = 42$;
- 4) если $|e \cap f| = 5$, то $|W(e) \cap W(f)| = 54$;
- 5) если $|e \cap f| = 6$, то $|W(e) \cap W(f)| = 104$.

Утверждение 2. Пусть в гиперграфе H есть три различных ребра e, f, h , для которых $|e \cap f| \notin \{4, 5\}$, $|e \cap h| \notin \{4, 5\}$. Тогда H обладает свойством B_3 .

Утверждение 3. Пусть в гиперграфе H есть два различных ребра e и f , для которых $|e \cap f| = 3$. Тогда H обладает свойством B_3 .

Утверждение 4. Пусть в гиперграфе H есть два различных ребра e и f , для которых $|e \cap f| = 6$. Тогда H обладает свойством B_3 .

Утверждение 5. Пусть в H никакие рёбра не имеют мощностей попарных пересечений 3 и 6. Пусть существуют рёбра $e, f \in E$, для которых $|e \cap f| = 5$. Тогда H обладает свойством B_3 .

Утверждение 6. Пусть в H никакие рёбра не имеют пересечения по 3 или 6 вершинам. Пусть не существует таких рёбер $e, f \in E$, что $|e \cap f| = 5$. Тогда H обладает свойством B_3 .

Ясно, что утверждениями 3—6 исчерпывается лемма 2.

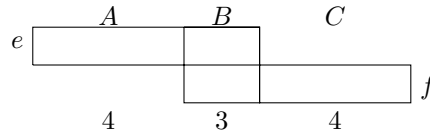
2.3.2. Доказательство утверждения 1

Доказательство пункта 1) очевидно.

Докажем пункт 2). Введём следующие обозначения:

$$A = e \setminus f, \quad B = e \cap f, \quad C = f \setminus e, \quad D = V \setminus (e \cup f).$$

Ясно, что $A \cup B \cup C \cup D = V$.

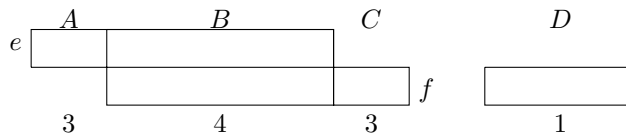


С точностью до переименования цветов половинные раскраски, в которых 3-плохими являются оба ребра, это

- а) $A = \{c, c, c, c\}, B = \{k, c, c\}, C = \{k, k, k, k\}$;
- б) $A = \{c, c, c, k\}, B = \{k, c, c\}, C = \{k, k, k, k\}$;
- в) $A = \{c, c, c, c\}, B = \{k, k, c\}, C = \{k, k, k, c\}$;
- г) $A = \{c, c, c, c\}, B = \{k, k, c\}, C = \{k, k, k, k\}$.

Раскрасок типа а) всего $C_3^1 \cdot 2$, раскрасок типа б) — $C_4^1 \cdot C_3^2 \cdot 2$, раскрасок типа в) — $C_4^1 \cdot C_3^2 \cdot 2$, типа г) — $C_3^1 \cdot 2$. Суммируя, получаем 60.

Докажем утверждение 3).

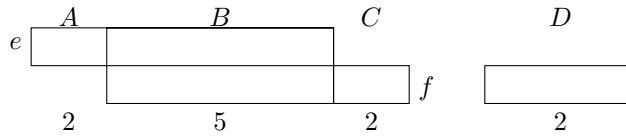


С точностью до переименования цветов половинные раскраски, в которых 3-плохими являются оба ребра, это

- а) $A = \{с, с, к\}, B = \{к, к, к, к\}, C = \{к, с, с\}, D = \{с\};$
- б) $A = \{с, с, с\}, B = \{к, к, с, с\}, C = \{к, к, к\}, D = \{к\};$
- в) $A = \{с, с, с\}, B = \{к, к, с, с\}, C = \{к, к, к\}, D = \{с\}.$

Раскрасок типа а) всего $C_3^1 \cdot C_3^1 \cdot 2$, раскрасок типа б) — $C_4^2 \cdot 2$, раскрасок типа в) — $C_4^2 \cdot 2$. Суммируя, получаем 42.

Докажем утверждение 4).

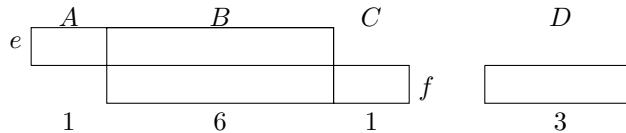


С точностью до переименования цветов половинные раскраски, в которых 3-плохими являются оба ребра, это

- а) $A = \{к, к\}, B = \{с, с, с, с, с\}, C = \{к, к\}, D = \{к, с\};$
- б) $A = \{к, к\}, B = \{с, с, с, с, с\}, C = \{к, к\}, D = \{к, к\};$
- в) $A = \{с, к\}, B = \{с, с, с, с, с\}, C = \{к, к\}, D = \{к, к\};$
- г) $A = \{к, к\}, B = \{с, с, с, с, с\}, C = \{к, с\}, D = \{к, к\};$
- д) $A = \{с, к\}, B = \{с, с, с, с, к\}, C = \{к, с\}, D = \{к, к\}.$

Раскрасок типа а) всего $C_2^1 \cdot 2$, типа б) всего 2, раскрасок типа в) — $C_2^1 \cdot 2$, раскрасок типа г) всего $C_2^1 \cdot 2$, типа д) — $C_5^4 \cdot C_2^1 \cdot C_2^1 \cdot 2$. Суммируя, получаем 54.

Докажем утверждение 5).



С точностью до переименования цветов половинные раскраски, в которых 3-плохими являются оба ребра, это

- а) $A = \{к\}, B = \{с, с, с, с, с, с\}, C = \{к\}, D = \{к, к, к\};$
- б) $A = \{с\}, B = \{с, с, с, с, с, к\}, C = \{к\}, D = \{к, к, к\};$
- в) $A = \{к\}, B = \{с, с, с, с, с, к\}, C = \{с\}, D = \{к, к, к\};$
- г) $A = \{к\}, B = \{с, с, с, с, с, к\}, C = \{к\}, D = \{к, к, к\};$
- д) $A = \{к\}, B = \{с, с, с, с, с, к\}, C = \{к\}, D = \{к, к, с\};$
- е) $A = \{с\}, B = \{с, с, с, с, к, к\}, C = \{с\}, D = \{к, к, к\}.$

Раскрасок типа а) всего 2, типа б) — $C_6^5 \cdot 2$, типа в) — $C_6^5 \cdot 2$, типа г) — $C_6^5 \cdot 2$, типа д) — $C_6^5 \cdot C_3^2 \cdot 2$, типа е) — $C_6^4 \cdot 2$. Суммируя, получаем 104.

2.3.3. Доказательство утверждения 2

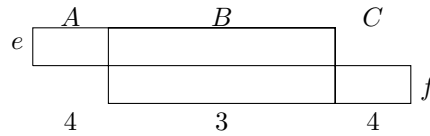
Пусть a_1, a_2 — различные рёбра H , отличные от e, f и h . Тогда из утверждения 1 получаем, что

$$\begin{aligned} |W(e) \cup W(f) \cup W(h) \cup W(a_1) \cup W(a_2)| &\leq \\ &\leq |W(e)| + |W(f) \setminus W(e)| + |W(h) \setminus W(e)| + |W(a_1) \setminus W(e)| + |W(a_2) \setminus W(e)| \leq \\ &\leq 224 + 164 + 164 + 182 \cdot 2 = 916 < 924. \end{aligned}$$

Следовательно, по замечанию 1 гиперграф H обладает свойством B_3 .

2.3.4. Доказательство утверждения 3

Согласно утверждению 2 достаточно рассмотреть случай, когда все оставшиеся рёбра a_1, a_2, a_3 имеют мощность пересечения 4 или 5 и с ребром f , и с ребром e . Введём следующие обозначения: $A = e \setminus f$, $B = e \cap f$, $C = f \setminus e$, $D = V \setminus (e \cup f)$.



Рассмотрим следующее множество раскрасок A_1 : в множествах A и C по две синие и по две красные вершины, в множестве B две синие и одна красная вершина. Ясно, что $|A_1| = 108$. В любой раскраске из A_1 рёбра e и f являются 3-хорошими. Для ребра a_i , $i = 1, 2, 3$, возможны ситуации, обозначенные в таблице буквами греческого алфавита $\alpha, \beta, \beta', \gamma$. Ребро имеет тип α , если оно пересекается с множеством A по двум вершинам, с множеством B — по трём вершинам, с множеством C — по двум вершинам. Остальные обозначения аналогичны.

	$ a_i \cap A $	$ a_i \cap B $	$ a_i \cap C $
α	2	3	2
β	3	2	2
β'	2	2	3
γ	3	1	3

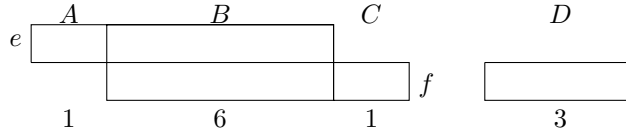
Легко убедиться, что $|W(a_i)| = 30$, если a_i типа α, β или β' , и $|W(a_i)| = 27$, если a_i типа γ . Таким образом,

$$\left| \bigcup_{e \in E} W(e) \right| \leq 30 \cdot 3 = 90 < 108,$$

следовательно, по замечанию 1 гиперграф H обладает свойством B_3 .

2.3.5. Доказательство утверждения 4

Введём следующие обозначения: $A = e \setminus f$, $B = e \cap f$, $C = f \setminus e$, $D = V \setminus (e \cup f)$.



Согласно утверждению 2 достаточно рассмотреть случай, когда все оставшиеся ребра a_1, a_2, a_3 имеют мощность пересечения 4 или 5 и с ребром f , и с ребром e . Для ребра a_i возможны следующие ситуации:

	$ a_i \cap A $	$ a_i \cap B $	$ a_i \cap C $	$ a_i \cap D $
α	1	4	1	1
β	1	4	0	2
β'	0	4	1	2
γ	0	5	0	2
δ	0	4	0	3
μ	1	3	1	2

Рассмотрим множество раскрасок A_2 : множество A синее, в B содержатся четыре красные и две синие вершины, C синее, в D содержатся две синие вершины и одна красная. Ясно, что $|A_2| = 45$. В любой раскраске из A_2 рёбра e и f являются 3-хорошими. Легко убедиться, что

- $|W(a_i)| = 13$, если a_i типа α ;
- $|W(a_i)| = 8$, если a_i типа β от β' ;
- $|W(a_i)| = 10$, если a_i типа γ ;
- $|W(a_i)| = 3$, если a_i типа δ ;
- $|W(a_i)| = 18$, если a_i типа μ .

Рассмотрим возникающие ситуации.

1. Рёбер типа μ менее двух. Тогда

$$\left| \bigcup_{e \in E} W(e) \right| \leq 13 \cdot 2 + 18 = 44 < 45,$$

следовательно, по замечанию 1 гиперграф H обладает свойством B_3 .

2. Рёбер типа μ хотя бы два, и третье ребро имеет тип, отличный от α, γ и μ . Тогда

$$\left| \bigcup_{e \in E} W(e) \right| \leq 13 \cdot 2 + 18 = 44 < 45,$$

следовательно, по замечанию 1 гиперграф H обладает свойством B_3 .

3. Рёбер типа μ хотя бы два, и третье ребро имеет тип α , γ или μ . Тогда искомой раскраской является следующая раскраска A_3 : множество A синее, в B содержатся три красные и три синие вершины, C синее, в D содержатся три красные вершины. Ясно, что $|A_3| = 20$, рёбра e и f являются хорошими в любой раскраске из A_3 . В этой раскраске

- $|W(a_i)| = 4$, если a_i типа α ;
- $|W(a_i)| = 10$, если a_i типа γ ;
- $|W(a_i)| = 2$, если a_i типа μ .

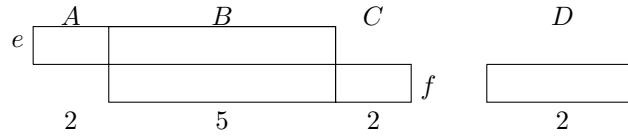
Таким образом,

$$\left| \bigcup_{e \in E} W(e) \right| \leq 2 \cdot 2 + 10 = 14 < 20.$$

Следовательно, по замечанию 1 гиперграф H обладает свойством B_3 .

2.3.6. Доказательство утверждения 5

Пусть $A = e \setminus f$, $B = e \cap f$, $C = f \setminus e$, $D = V \setminus (e \cup f)$.



Для оставшихся рёбер a_1, a_2, a_3 возможны девять различных вариантов распределения вершин по множествам A, B, C, D , они обозначены в таблице буквами греческого алфавита (пересекаться с e или f по не более чем двум вершинам оставшиеся рёбра не могут из-за количества вершин).

Рассмотрим следующее множество A_4 раскрасок вершин H : в каждом из множеств A, C, D содержатся одна красная и одна синяя вершина, в B — две красные и три синие вершины. Значения $|W(a_i)|$ для рёбер a_i также приведены в таблице.

	$ a_i \cap A $	$ a_i \cap B $	$ a_i \cap C $	$ a_i \cap D $	$ W(a_i) $
α	2	3	2	0	8
β	2	2	2	1	16
γ	1	3	2	1	24
γ'	2	3	1	1	24
γ''	1	3	1	2	24
θ	0	4	1	2	16
θ'	1	4	0	2	16
η	0	5	0	2	0
ζ	1	4	1	1	28

Ясно, что $|A_4| = 80$, рёбра e и f являются хорошими в любой раскраске из A_4 . Рассмотрим возникающие ситуации.

1. Рёбер типа ζ не более одного. Тогда

$$\left| \bigcup_{e \in E} W(e) \right| \leq 24 \cdot 2 + 28 = 76 < 80.$$

По замечанию 1 гиперграф H обладает свойством B_3 .

2. Рёбер типа ζ два. Обозначим их a_1, a_2 . Тогда необходимо рассмотреть другую раскраску, если последнее ребро a_3 имеет тип γ, γ' или γ'' . Рассмотрим следующее множество A_5 раскрасок вершин H . В множестве D содержатся одна синяя и одна красная вершина, в каждом из множеств A, C содержатся две синие вершины, в B — четыре красные и одна синяя вершина. Ясно, что $|A_5| = 10$. При этом $W(e) = W(f) = \emptyset, W(a_i) = 1, i = 1, 2$. Для случая γ или γ' существует три раскраски, в которых последнее ребро является плохим, для случая γ'' не существует таких раскрасок.

Таким образом,

$$\left| \bigcup_{e \in E} W(e) \right| \leq 3 \cdot 2 + 1 = 7 < 10,$$

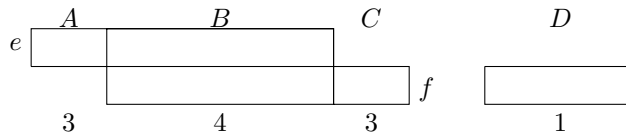
следовательно, по замечанию 1 гиперграф H обладает свойством B_3 .

3. Рёбер типа ζ три. Обозначим их a_1, a_2, a_3 . Рассмотрим следующее множество A_6 раскрасок вершин H : в каждом из множеств A, C, D содержатся две синие вершины, в B — четыре красные и одна синяя вершина. Ясно, что $|A_6| = 5$. При этом $W(e) = W(f) = W(a_i) = \text{varnothing}, i = 1, 2, 3$, т. е. гиперграф H обладает свойством B_3 .

2.3.7. Доказательство утверждения 6

Из условия следует, что для любых $e, f \in E$ выполнено $|e \cap f| = 4$, так как иначе какие-то рёбра либо не пересекались, либо пересекались бы по одной или двум вершинам, что невозможно из-за количества вершин.

Пусть $A = e \setminus f, B = e \cap f, C = f \setminus e, D = V \setminus (e \cup f)$.



Для ребра a_i возможны следующие ситуации:

	$ a_i \cap A $	$ a_i \cap B $	$ a_i \cap C $	$ a_i \cap D $
α	2	2	2	1
β	3	1	3	0

Рассмотрим множество раскрасок A_2 : множество A содержит две красные и одну синюю вершины, в B содержатся три синие и одна красная вершины, C содержит две красные и одну синюю вершины, D синее. Ясно, что $|A_2| = 36$. В любой раскраске из A_2 рёбра e и f являются 3-хорошими. Легко убедиться, что

- $|W(a_i)| = 10$, если a_i имеет тип α ;
- $|W(a_i)| = 9$, если a_i имеет тип β .

Таким образом,

$$\left| \bigcup_{e \in E} W(e) \right| \leq 10 \cdot 3 = 30 < 36,$$

следовательно, по замечанию 1 гиперграф H обладает свойством B_3 .

Доказав эти утверждения, мы рассмотрели все возможные варианты пересечения рёбер гиперграфа H . Следовательно, лемма 2 доказана.

2.4. Доказательство леммы 3

Докажем лемму индукцией по числу вершин гиперграфа H .

БАЗА ИНДУКЦИИ. Пусть $|V| = 11$. Тогда гиперграф H обладает свойством B_3 по лемме 2.

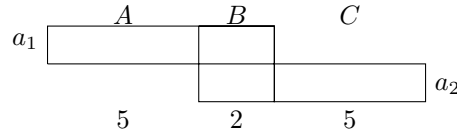
ИНДУКЦИОННЫЙ ПЕРЕХОД. Предположим, что утверждение леммы верно для $|V| = n$, $n \geq 11$. Докажем, что оно верно для $|V| = n + 1$. Заметим, что при $|V| \geq 12$ в гиперграфе H найдётся либо вершина степени 1, либо вершина степени 2. Рассмотрим два этих случая.

Случай 1. Пусть $A \in V$ — вершина степени 1, $a \in E$ — ребро, содержащее A . Рассмотрим 7-однородный гиперграф H' , полученный из H следующим образом: $V' = V \setminus \{A\}$, $E' = (E \setminus \{a\}) \cup \{a'\}$, где $a' = (a \setminus \{A\}) \cup \{B\}$, B — любая вершина из $V \setminus a$. Тогда $|V'| = n$, $|E'| = 5$, значит, граф H' обладает свойством B_3 по предположению индукции, т. е. существует раскраска множества V' , при которой все его ребра 3-хорошие. Применим к $V' = V \setminus \{A\}$ ту же раскраску, а вершине A присвоим цвет вершины B в V' . В описанной раскраске все рёбра H оказываются 3-хорошими.

Случай 2. Пусть $A \in V$ — вершина степени 2, $a_1, a_2 \in E$ — рёбра, содержащие A , множество $V \setminus (a_1 \cup a_2)$ не пусто. Рассмотрим 7-однородный гиперграф $H' = (V', E')$, где $V' = V \setminus \{A\}$, $E' = (E \setminus (\{a_1\} \cup \{a_2\})) \cup \{a'_1\} \cup \{a'_2\}$, $a'_1 = (a_1 \setminus \{A\}) \cup \{B\}$, $a'_2 = (a_2 \setminus \{A\}) \cup \{B\}$, а B — вершина из $V \setminus (a_1 \cup a_2)$. Ясно, что $|V'| = n$, $|E'| = 5$. Значит, по предположению существует такая раскраска V' , что все рёбра H' хорошие. Применяем теперь эту раскраску к $V \setminus \{A\}$, а вершину A красим в тот цвет, который имела вершина B в H' . Получаем раскраску гиперграфа H , в которой все его рёбра хорошие.

Случай 3. Пусть $S \in V$ — вершина степени 2, $a_1, a_2 \in E$ — рёбра, содержащие S , множество $V \setminus (a_1 \cup a_2)$ пусто. Это возможно в двух случаях: $|V| = 12$, $a_1 \cap a_2 = 2$ и $|V| = 13$, $a_1 \cap a_2 = 1$.

СЛУЧАЙ 3.1. Введём следующие обозначения: $A = a_1 \setminus a_2$, $B = a_1 \cap a_2$, $C = a_2 \setminus a_1$. Ясно, что $A \cup B \cup C = V$.



Рассмотрим множество раскрасок A_1 : в множестве A три красные и две синие вершины, в B одна вершина синяя и одна красная, в C три синие и две красные вершины. Ясно, что $|A_1| = 200$. В любой раскраске из A_1 рёбра a_1 и a_2 являются 3-хорошими.

Для оставшихся рёбер a_3, a_4, a_5 возможны девять различных вариантов распределения вершин по множествам A, B, C , они обозначены в таблице буквами греческого алфавита (пересекаться с B по двум вершинам оставшиеся рёбра не могут). Значения $|W(a_i)|$ для рёбер a_i приведены в таблице.

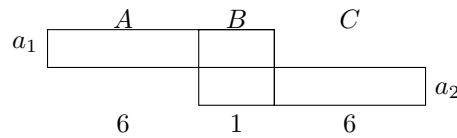
	$ a_i \cap A $	$ a_i \cap B $	$ a_i \cap C $	$ W(a_i) $
α	5	1	1	40
β	5	0	2	20
γ	4	1	2	56
δ	4	0	3	36
μ	3	1	3	60
δ'	3	0	4	36
γ'	2	1	4	56
β'	2	0	5	20
α'	1	1	5	40

Тогда

$$\left| \bigcup_{e \in E} W(e) \right| \leq 3 \cdot 60 = 180 < 200.$$

По замечанию 1 гиперграф H обладает свойством B_3 .

СЛУЧАЙ 3.2. Воспользуемся обозначениями из случая 3.1.



Рассмотрим множество раскрасок A_2 : в множестве A три красные и три синие вершины, множество B синее, в C три синие и три красные вершины. Ясно, что $|A_2| = 400$. В любой раскраске из A_2 рёбра a_1 и a_2 являются 3-хорошими.

Значения $|W(a_i)|$ для рёбер a_i , $i = 3, 4, 5$, приведены в таблице.

	$ a_i \cap A $	$ a_i \cap B $	$ a_i \cap C $	$ W(a_i) $
α	6	0	1	0
β	5	0	2	80
γ	4	0	3	104
γ'	3	0	4	104
β	2	0	5	80
α	1	0	6	0

Тогда

$$\left| \bigcup_{e \in E} W(e) \right| \leq 3 \cdot 104 < 400.$$

По замечанию 1 гиперграф H обладает свойством B_3 .

Лемма доказана. Теорема 1 доказана.

3. Доказательство теоремы 2

Для доказательства теоремы 2 нам достаточно обосновать следующие леммы.

Лемма 4. Имеет место оценка $m_4(8) \leq 4$.

Лемма 5. Пусть 8-однородный гиперграф $H = (V, E)$ таков, что $|V| = 9$, $|E| = 3$. Тогда H обладает свойством B_4 .

Лемма 6. Пусть 8-однородный гиперграф $H = (V, E)$ таков, что $|V| \geq 10$, $|E| = 3$. Тогда H обладает свойством B_4 .

3.1. Доказательство леммы 4

Рассмотрим 8-однородный гиперграф

$$H = (V, E), \quad V = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}, \quad E = \{e_1, e_2, e_3, e_4\},$$

где

$$\begin{aligned} e_1 &= \{1, 2, 3, 4, 5, 6, 7, 8\}, & e_2 &= \{1, 2, 3, 4, 5, 9, 10, 11\}, \\ e_3 &= \{1, 2, 6, 7, 8, 9, 10, 11\}, & e_4 &= \{3, 4, 5, 6, 7, 8, 9, 10\}. \end{aligned}$$

Назовём два цвета, в которые покрашены вершины нашего графа, красным и синим. Покажем, что H не обладает свойством B_4 . Без ограничения общности вершина 11 красная. Среди вершин $\{1, 2, 9, 10\}$ хотя бы одна синяя, иначе во втором ребре пять красных. Пусть это вершина 1. Тогда если среди вершин $\{2, 9, 10\}$ хотя бы две красные, то среди $\{3, 4, 5\}$ хотя бы две синие и среди $\{6, 7, 8\}$ хотя бы две синие. Но тогда в первом ребре пять синих вершин. Поэтому среди $\{2, 9, 10\}$ по крайней мере две синие, пусть это вершины 2 и 9.

Среди вершин $\{3, 4, 5\}$ хотя бы две красные, иначе во втором ребре пять синих. Пусть это вершины 3 и 4. Аналогично среди вершин $\{6, 7, 8\}$ хотя бы две красные, иначе в третьем ребре пять синих. Пусть это вершины 6 и 7. Тогда, проводя рассуждения для первого ребра, получаем, что вершины 5, 8 синие. Из рассуждений для второго ребра следует, что вершина 10 красная. Но в этом случае в четвёртом ребре пять красных вершин, противоречие. Остальные случаи рассматриваются аналогично.

3.2. Доказательство леммы 5

Пусть 8-однородный гиперграф $H = (V, E)$ таков, что $|E| = 3$. Тогда $|V| \geq 9$. Приведём пример раскраски, в которой все рёбра являются 4-хорошими.

g	к	с,с,с,с,к,к		к
e	к	с,с,с,с,к,к	к	
f		с,с,с,с,к,к	к	к

3.3. Доказательство леммы 6

Докажем лемму индукцией по числу вершин гиперграфа H .

База индукции. Пусть $|V| = 9$. Тогда H обладает свойством B_4 по лемме 5.

ИНДУКЦИОННЫЙ ПЕРЕХОД. Предположим, что утверждение леммы верно для $|V| = n$, $n \geq 9$. Докажем, что оно верно для $|V| = n + 1$. Рассмотрим два случая.

СЛУЧАЙ 1. Пусть существует A — вершина степени 1 из V , $a \in E$ — ребро, содержащее A . Рассмотрим 8-однородный гиперграф H' , полученный из H следующим образом: $V' = V \setminus \{A\}$, $E' = (E \setminus \{a\}) \cup \{a'\}$, где $a' = (a \setminus \{A\}) \cup \{B\}$, B — любая вершина из $V \setminus a$. Тогда $|V'| = n$, $|E'| = 3$, значит, граф H' обладает свойством B_4 по предположению индукции, т. е. существует раскраска V' , при которой все его рёбра хорошие. Применим к $V' = V \setminus \{A\}$ ту же раскраску, а вершине A присвоим цвет вершины B в V' . Таким образом, все рёбра H оказываются 4-хорошими в описанной раскраске.

СЛУЧАЙ 2. Введём следующие обозначения: $A = g \cap e$, $B = e \cap f \cap g$, $C = f \cap e$, and $D = g \cap f$.

При $|V| \geq 10$ в случае, когда гиперграф не содержит вершин степени 1, имеет место одна из следующих ситуаций.

	A	B	C	D
g	2	4		2
e	2	4	2	
f		4	2	2

g	3	2		3
e	3	2	3	
f		2	3	3

g	4	0		4
e	4	0	4	
f		0	4	4

Для каждой из них, очевидно, существует раскраска, в которой каждое из рёбер является 4-хорошим. Лемма доказана. Таким образом, теорема 2 доказана.

4. Описание работы алгоритма

Для дальнейшего нам потребуется следующее определение.

Определение 1. *Расстоянием от вектора a до вектора b назовём количество компонент, в которых вектор a содержит 1, а вектор b содержит 0.*

Вначале занумеруем вершины графа числами от 0 до $n - 1$. Параметр `count` задаёт число вершин в одном ребре. С помощью функции `make_graph` сформируем двумерный массив всевозможных рёбер `rebra` размерности $[kol, count]$, где $kol = C_n^k$, а строки состоят из всевозможных подмножеств длины n из множества номеров вершин. В функции `make_rask` заполняем двумерный массив `rask` размерности $[kol, k_1]$, где $k_1 = 2^n$ — число всевозможных раскрасок множества из n вершин в два цвета, в котором для всевозможных раскрасок всех вершин указывается, является ли каждое ребро плохим (1) или хорошим (0) в данной раскраске. Эта проверка выполняется при помощи функции `check`.

Алгоритм `check`

```

Data:  $k$ 
Result: алгоритм check
1  инициализация;
2  if количество вершин одного цвета  $< k$  then
3      ребро плохое;
4      количество плохих рёбер  $+= 1$ ;
5  else
6      ребро хорошее;
7  end

```

Если функция `make_rask` не нашла плохих рёбер, то это означает, что существует двухцветная раскраска множества вершин гиперграфа, в которой все рёбра являются k -хорошими, т. е. граф с таким количеством вершин при любом количестве рёбер обладает свойством B_k .

Если функция `make_rask` нашла плохие рёбра в каждой раскраске, то вызывается функция `make_set`, которая выбирает минимальное количество рёбер, достаточное для того, чтобы при любой раскраске среди выбранных рёбер было хотя бы одно k -плохое. Заведём итоговый вектор — вектор, имеющий длину, равную количеству всевозможных раскрасок, и на каждом шаге будем стремиться максимально уменьшить число нулей в итоговом векторе, где ноль означает, что плохих рёбер для данной раскраски ещё нет, а единица — что есть. При этом первое ребро можно выбрать произвольным образом, так как изначально все вершины симметричны; выбираем нулевое ребро, берём нулевую строку массива `rask`, приравниваем итоговый вектор к этой строке. Следующим добавляем ребро, для которого расстояние от соответствующей ему строки матрицы `rask` до итогового вектора максимально, производим покомпонентную дизъюнкцию этой строки с итоговым вектором. Алгоритм конечен, так как если мы добавим все рёбра, то в соответствии с результатом работы функции `make_rask` плохие рёбра будут в каждой раскраске.

Алгоритм `make_set`

Data: матрица <code>rask</code> Result: алгоритм <code>make_set</code> 1 инициализация; 2 добавляем нулевой вектор в набор, приравниваем итоговый вектор нулевой строке матрицы <code>rask</code>; 3 while <i>итоговый вектор не заполнен единицами</i> do 4 добавляем в набор вектор, для которого расстояние от соответствующей ему строки матрицы <code>rask</code> до итогового вектора максимально 5 формируем итоговый вектор: производим покомпонентную дизъюнкцию строки матрицы <code>rask</code> с итоговым вектором; 6 end

В некоторых случаях удаётся уменьшить набор рёбер, если, выбросив ребро и выполнив проверку при помощи функции `check_var`, мы снова получим набор рёбер, удовлетворяющий условию: итоговый вектор заполнен единицами.

Отметим, что программа легко обобщается для случая, когда раскраска производится в r различных цветов.

5. Полный текст программы

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;
```

```
namespace Two_colors
{
    class Program
    {
        static int n;           // число вершин
        static int count = 6;    // число вершин в одном
                                // ребре
        static int kol;         // число рёбер
                                // по count вершин
        static int[, ] rebra;    // матрица всевозможных
                                // рёбер
        static int parl = 0;     // это число k-1
                                // из начальной задачи
        static int kl;           // число раскрасок
                                // при n вершинах 2**n
        static int[, ] rask;     // матрица плохих (1) и
                                // хороших (0) рёбер при
                                // всевозможных раскрасках
        static int[] result_var; // массив номеров
                                // выбранных рёбер

        static void Main(string[] args)
        {
            // n = 10;
            for (n=count+1; n<=20; n++)
            {
                Console.WriteLine();
                Console.WriteLine("Количество вершин: "+n);
                kol = (int)(factor(n) /
                    (factor(count) * factor(n - count)));
                rebra = new int[kol, count];
                make_graph();

                if (make_rask())
                {
                    make_set();
                    Console.Write("Пожалуйста, нажмите пробел,");
                    Console.Write(" чтобы продолжить");
                    Console.WriteLine();
                    Console.ReadKey();
                    check_var();
                }
            }
        }
    }
}
```

```
        Console.ReadKey();
    }

    static bool make_rask()
    {
        int[] bad = new int[kol];
        int[] count_bad = new int[kol];
        int[] distance = new int[kol];
        int[,] coloring = new int[kol, count];
        int[] color1 = new int[n];
        k1 = 1;
        for (int i = 0; i < n-1; i++)
            k1 *= 2;
        rask = new int[kol, k1];
        for (int q = 0; q < kol; q++)
        {
            for (int p = 0; p < k1; p++)
            {
                rask[q, p] = 0;
            }
        }

        int m = -1;
        for (int k2 = 0; k2 < k1; k2++)
        {
            m++;
            if (check(color1, kol, rebra,
                bad, count_bad, coloring) == true)
            {
                Console.WriteLine
                    ("Все ребра хорошие в одной из раскрасок");
                return false;
            }
            else // для каждой раскраски
                // существует ребро, которое в ней плохое
            {
                for (int k = 0; k < kol; k++)
                {
                    if (bad[k] == 1)
                        rask[k, m] = 1;
                    else rask[k, m] = 0;
                }
            }
        }
    }
}
```

```
for (int k3 = n - 1; k3 >= 0; k3--)  
{  
    if (color1[k3] < 1)  
    {  
        color1[k3]++;  
        for (int l = k3 + 1; l < n; l++)  
        {  
            color1[l] = 0;  
        }  
        break;  
    }  
}  
return true;  
}  
  
static void make_set()  
{  
    int[] itog = new int[k1];    // суммарный вектор,  
                                // в котором единицы для  
                                // всех выбранных рёбер  
    int[] result_rebra = new int[kol]; // массив,  
                                        // где единицами  
                                        // обозначены  
                                        // выбранные ребра  
  
    int d = 0;  
    int sum_itog = 0;  
  
    while (!itog_check(itog)) // пока itog заполнится 1  
    {  
        d = dist(itog, kol, rask);  
        sum_itog =  
            new_coloring(rask, d, itog, result_rebra);  
    }  
  
    int sum = 0;  
    for (int y = 0; y < kol; y++)  
    {  
        if (result_rebra[y] == 1)  
        {  
            sum++;  
        }  
    }  
    Console.WriteLine("Количество ребер = " + sum);  
}
```

```
result_var = new int[sum];
int i = 0;
for (int y = 0; y < kol; y++)
{
    if (result_rebra[y] == 1)
    {
        result_var[i]=y;
        i++;
    }
}

static void check_var()
{
    // проверка варианта
    int kol_st = result_var.Length;
    int[] vect = new int[kol_st];

    int sum = 0;
    int i, i2, i3;

    for (i = 0; i < result_var.Length; i++)
        vect[i] = result_var[i];
    print(vect);

    for (i3 = 0; i3 < k1; i3++)
    {
        sum = 0;
        for (i2 = 0; i2 < kol_st; i2++)
            sum += rask[vect[i2], i3];

        if (sum == 0)
        {
            for (i2 = 0; i2 < kol_st; i2++)
                Console.Write(rask[vect[i2], i3] + " ");
            Console.WriteLine();
            Console.WriteLine("NO!");
            Console.ReadKey();
            return;
        }
    }

    Console.WriteLine("OK!");
    Console.WriteLine("Пёбра:");
```

```
for (i2 = 0; i2 < kol_st; i2++)
{
    for (int j = 0; j < count; j++)
        if (rebra[vect[i2], j] < 10)
            Console.Write(" "+rebra[vect[i2], j] + " ");
        else
            Console.Write(rebra[vect[i2], j] + " ");
    Console.WriteLine();
}

static void check_var(int k)
{
    // проверка варианта
    int kol_st = result_var.Length-k;
    int[] vect = new int[kol_st];

    int sum = 0;
    int j;
    bool flag = true;
    for (int m = 0; m < result_var.Length; m++)
    {
        j = 0;
        for (int i = 0; i < result_var.Length; i++)
        {
            if (i != m)
            {
                vect[j] = result_var[i];
                j++;
            }
        }
        Console.WriteLine("Проверка, нет ли лишних рёбер");
        print(vect);
        flag = true;
        for (int i3 = 0; i3 < k1; i3++)
        {
            sum = 0;
            for (int i2 = 0; i2 < kol_st; i2++)
                sum += rask[vect[i2], i3];
            if (sum == 0)
            {
                flag = false;
            }
        }
    }
}
```

```
        if (flag)
        {
            Console.WriteLine();
            Console.WriteLine("YES!");
        }
    }

    Console.WriteLine("END!");
}

static ulong factor(int m)
{
    if (m == 1) return 1;
    return (ulong)m * factor(m - 1);
}

static bool itog_check(int[] itog)
{
    int k = 0;
    for (k = 0; k < itog.Length; k++)
        if (itog[k] == 0)
            return false;
    return true;
}

static int new_coloring(int[,] rask, int num,
    int[] itog, int[] result_rebra)
{
    for (int i = 0; i < itog.Length; i++)
        if (rask[num, i] == 1) itog[i] = 1;

    int sum = 0;
    for (int i = 0; i < itog.Length; i++)
        if (itog[i] == 1) sum++;
    result_rebra[num] = 1;
    return sum;
}

static int dist(int[] itog, int kol, int[,] rask)
{
    int i = 0;
    int i_max = 0;
    int[] distance = new int[kol];
```

```
for (i = 0; i < kol; i++)
{
    for (int j = 0; j < itog.Length; j++)
    {
        if (itog[j] == 0 && rask[i, j] == 1)
            distance[i]++;
    }
}
int max = distance[0];
for (i = 0; i < kol; i++)
{
    if (distance[i] > max)
    {
        max = distance[i]; i_max = i;
    }
}
return i_max;
}

static void make_graph()
{
    int[] v = new int[count];
    for (int i = 0; i < count; i++)
        v[i] = i + 1;

    for (int i = 0; i < kol; i++)
    {
        for (int j = 0; j < count; j++)
            rebra[i, j] = v[j];
        for (int k = count-1; k >= 0; k--)
        {
            if (v[k] < n - (count-1 - k))
            {
                v[k]++;
                for (int l = k + 1; l < count; l++)
                {
                    v[l] = v[k] + l - k;
                }
                break;
            }
        }
    }
}
```

```
static bool check(int[] color1, int kol, int[,] rebra,
    int[] bad, int[] count_bad, int[,] coloring)
{
    int sum = 0;
    bool res = true;

    for (int i = 0; i < kol; i++)
    {
        bad[i] = 0;
        sum = 0;
        for (int j = 0; j < count; j++)
        {
            coloring[i, j] = color1[rebra[i, j] - 1];
            sum += coloring[i, j];
        }
        if (sum <= par1 || sum >= count-par1)
        {
            bad[i] = 1;
            count_bad[i]++;
            res = false;
        }
    }
    return res;
}

static void print(int[] mas)
{
    for (int i = 0; i < mas.Length; i++)
    {
        Console.Write(mas[i]);
        Console.Write(' ');
    }
    Console.WriteLine();
}
}
```

Литература

- [1] Райгородский А. М., Шабанов Д. А. Задача Эрдёша—Хайнала о раскрасках гиперграфов, её обобщения и смежные проблемы // Успехи мат. наук. — 2011. — Т. 66, № 5 (401). — С. 109—182.

- [2] Розовская А. П. О двухцветных раскрасках общего вида для равномерных гиперграфов // Докл. РАН. — 2009. — Т. 429, № 3. — С. 309—311.
- [3] Розовская А. П., Титова М. В., Шабанов Д. А. О половинных раскрасках гиперграфов // Фундамент. и прикл. мат. — 2009. — Т. 15, вып. 7. — С. 141—163.
- [4] Тепляков С. М. Рекуррентные верхние оценки в задаче Эрдёша—Хайнала о раскраске гиперграфа и в её обобщениях // Тр. МФТИ. — 2012. — Т. 4, № 1 — С. 141—150.
- [5] Тепляков С. М. Верхняя оценка в задаче Эрдёша—Хайнала о раскраске гиперграфа // Мат. заметки. — 2013. — Т. 93, № 1. — С. 148—151.
- [6] Черкашин Д. Д., Куликов А. Б. О двухцветных раскрасках гиперграфов // Докл. РАН. — 2011. — Т. 463, № 3. — С. 316—319.
- [7] Шабанов Д. А. Об одной комбинаторной задаче Эрдёша // Докл. РАН. — 2004. — Т. 396, № 2. — С. 166—169.
- [8] Шабанов Д. А. Экстремальные задачи для раскрасок равномерных гиперграфов // Изв. РАН. Сер. мат. — 2007. — Т. 71, № 6. — С. 183—222.
- [9] Шабанов Д. А. Рандомизированные алгоритмы раскрасок гиперграфов // Мат. сб. — 2008. — Т. 199, № 7. — С. 139—160.
- [10] Abbott H. L., Hanson D. On a combinatorial problem of Erdős // Can. Math. Bull. — 1969. — Vol. 12. — P. 823—829.
- [11] Abbott H. L., Hare D. R. Families of 4-sets without property B // Ars Combinatoria. — 2001. — Vol. 60. — P. 239—245.
- [12] Abbott H. L., Moser L. On a combinatorial problem of Erdős and Hajnal // Can. Math. Bull. — 1964. — Vol. 7. — P. 177—181.
- [13] Erdős P., Hajnal A. On a property of families of sets // Acta Math. Acad. Sci. Hungar. — 1961. — Vol. 12. — P. 87—123.

