ГУО «Боровлянская гимназия»

*Исследовательская работа*

# Характеристика таблиц

Авторы: Подгайский Артем
Учащийся 11 класса
Руководитель: Шабан Алла Брониславовна

# Содержание

# Введение

В математике часто можно наблюдать, как простые на первый взгляд задачи, некоторые факты в которых кажутся совершенно очевидными, имеют непростые решения, которые используют факты из тех разделов математики, которые, казалось бы, никак не связаны с решаемой задачей.

На XXIV республиканском турнире юных математиков была предложена задача "Характеристика таблиц", в которой предложено изучить некоторые свойства определенного вида таблиц (*Рис. 1*) .

| | | | | |
|---|---|---|---|---|
| + | + | - | + | - |
| - | - | - | + | + |
| + | - | - | + | - |

| | | | | |
|---|---|---|---|---|
| + | - | + | + | + |
| - | + | + | + | - |
| + | + | + | + | + |

Рис. 1: Прямоугольные таблицы $3 \times 5$, в которые расставлены знак «+» или «-»

В рассматриваемые прямоугольные таблицы вписаны знаки «+» или «-». Для столбца или строки таблицы определена *операция смены знаков*, которая изменяет знаки в столбце или строке на противоположные. Например на *Рис. 1* во второй таблице операция смены знаков произведена в столбцах 2, 3, 5 относительно первой таблицы.

**Определение.** ***Характеристикой заданной таблицы*** *называется минимальное количество минусов, которое можно получить для данной таблицы, применяя операцию смены знаков к любым столбцам и строкам любое количество раз.*

**Объект исследования:** прямоугольные таблицы с вписанными в них знаками «+» и «-», в которых разрешается производить смену знаков.

**Цель исследования:** построить математическую модель рассматриваемых таблицы, изучить свойства характеристик таблиц.

**Задачи исследования:**

- Нахождение всех возможных характеристик таблиц $2\times n$, $3\times n$, и оценка диапазона характеристик таблиц других размерностей.
- Создание алгоритма нахождения характеристики заданной таблицы любой размерности.
- Нахождение характеристик таблиц любой размерности, в которых знаки расположены определенным образом.

Интерес в данной задаче представляют не только математически обоснованные факты, но и оценки, полученные, например, с помощью компьютерной программы.

# 1 Построение математической модели таблиц

Строки и столбцы таблицы будем называть *рядами* таблицы.

Для решения задачи, заменим в таблице плюсы и минусы на числа "1" и "-1" соответственно. *Тогда заметим, что смена знаков эквивалентна домножению каждого числа в ряду на -1.* Далее мы часто будем использовать как «+» и «-» для удобства, так и «1» и «-1» для вычислений.

Характеристику заданной таблицы будем обозначать $w$. Множество харакеристик таблицы $m \times n$ обозначим как $W_{m\times n}$

**Определение.** ***Параметром ряда таблицы*** *будем называть величину, равную произведению всех чисел в данном ряду и обозначать $\varepsilon_i$ (для столбца – $\varepsilon_{x,i}$, для строки – $\varepsilon_{y,i}$).*

Ясно, что параметр ряда может принимать значения 1 и -1.

Если таблицу $B$ можно получить из таблицы $A$ путем применения к рядам $B$ операции смены знаков несколько раз, такое отношение будем записывать как $A \sim B$. Вполне очевидно, что такое отношение таблиц является отношением эквивалентности[2]:

1. Рефлексивность: $A \sim A$. Таблицу $A$ можно получить из таблицы $A$, не производя смену знаков.

2. Симметричность: если $A \sim B$, то $B \sim A$. Если таблицу $B$ можно получить из $A$, то для того, чтобы получить $A$ из $B$, необходимо применить операцию смены знаков к тем же рядам, но в обратном порядке.

3. Транзитивность: если $A \sim B$ и $B \sim C$, то $A \sim C$. Очевидно, что если существует способ получить из $A$ таблицу $B$, и из $B$ – $C$, тогда чтобы получить из $A$ таблицу $C$, достаточно сначала получить $B$ из $A$, а потом $C$ из $B$.

**Определение.** *Классом таблиц будем называть множество, содержащее в себе все таблицы, каждая из которых может быть получена из любой другой таблицы в этом множестве путем применения операции смены знаков в произвольных рядах один или несколько раз.*

Ясно, что класс таблиц является классом эквивалентности. Тогда, согласно определению характеристики таблицы, характеристики таблиц одного класса равны.

**Лемма 1.** *Если длина ряда четна, то его параметр не меняется с изменением знаков в нем.*

*Доказательство.* Пусть в ряду с четной длиной содержится $n$ "-1" и $k$ "1". Ясно что длина ряда равна $n+k$ и, ввиду четности $n+k$, числа $n \equiv k \pmod 2$. Его параметр

$$\varepsilon_1 = (-1)^n \cdot 1^k = -nk$$

После смены знаков в ряду - количество "-1" - $k$, количество "1" - $n$ и параметр становится равным

$$\varepsilon_2 = (-1)^k \cdot 1^n$$

Поскольку $n \equiv k \pmod 2$, то $(-1)^n = (-1)^k$, откуда следует $\varepsilon_1 = \varepsilon_2$. □

**Лемма 2.** *При замене знаков в строке таблицы $m \times n$, каждый из параметров столбцов этой таблицы меняется на проитвоположный*

*Доказательство.* Заметим, что при замене знаков в строке, ровно одно число в каждом столбце меняется на противоположное. Пусть в столбце $i$ записаны числа $a_1, a_2, ..., a_m$. Его параметр равен $\varepsilon_i = a_1a_2...a_m$. Тогда при смене знаков в строке $j$, в этом столбце число $a_j$ изменяется на $-a_j$. Соответствнно параметр столбца становится равным $\varepsilon_i = a_1a_2...(-a_j)...a_n = -a_1a_2...a_n$, т.е изменяется на противоположный. А так как при смене строк в каждом столбце на противоположное меняется число под номером $j$, тогда и параметр каждого столбца меняется на противоположный. □

## 2 Все возможные характеристики таблицы $2 \times n$

Для таблицы $2 \times n$ введем следующее определение.

**Определение.** *Таблицу $2 \times n$ будем называть* ***правильной*** *если в ней в каждом столбце либо ровно один минус, либо 2 плюса.*

Очевидно, любую таблицу $2 \times n$ можно привести к правильному виду путем изменения знаков в столбцах. Если в каком-то столбце ровно один минус или два плюса, мы не меняем в нем знаки. Если же в каком-то столбце два минуса, применив операцию, получим в нем два плюса (на *рис. 2* обозначены все возможные расположения знаков в столбце длиной 2). Поэтому далее мы будем рассматривать только характеристики правильных таблиц, так как характеристика любой таблицы равна харакеристике соответствующей ей правильно таблице.

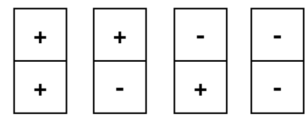

Рис. 2:

Определим множество параметров столбцов таблицы

$$E = \{\varepsilon_{x,1}, \varepsilon_{x,2}, ..., \varepsilon_{x,n}\}$$

Так как столбцы имеют четную длину, то по *Лемме 1* при изменении знаков в них, их параметры остаются постоянными. А по *Лемме 2* при изменении знаков в строках, параметры столбцов меняются на противоположные. Таким образом, как бы мы не меняли знаки в таблице, множество $E$ принимает только два значения.

$$E_1 = \{\varepsilon_{x,1}, \varepsilon_{x,2}, ..., \varepsilon_{x,n}\}$$

$$E_2 = \{-\varepsilon_{x,1}, -\varepsilon_{x,2}, ..., -\varepsilon_{x,n}\}$$

Далее заметим, что в правильной таблице для столбца длиной два, его $\varepsilon_i = 1$ тогда и только тогда, когда в нем отсутствуют "-1". И его $\varepsilon_i = -1$ тогда и только тогда, когда в нем ровно одна "-1" (легко это показать: для правильной таблицы 1 и -1 представимы в виде произведения двух чисел единственным образом: $1 = 1 \cdot 1$; $-1 = 1 \cdot (-1)$).

Значит количество "-1" в правильной таблице равно количество "-1" в множестве $E$, и также принимает только 2 значения. Тогда *характеристикой таблицы* $2 \times n$ будет являться минимальное из этих двух значений.

Обозначим количество минусов в множестве $E$ за $x_E$. Тогда $x_E$ может принимать любые значения $x_E \in [0; n]$. Если в множестве $E_1$ количество минусов $x_{E_1} \geq \frac{n}{2}$, тогда в множестве $E_2$ количество минусов $x_{E_2} = n - x_{E_1} \leq \frac{n}{2}$. Значит среди $x_{E_1}, x_{E_2}$ точно найдется число, меньшее чем $[\frac{n}{2}]$, и характеристика таблицы $2 \times n$ $w \leq [\frac{n}{2}]$

Если расположить в первой строчке $0 \geq k \leq [\frac{n}{2}]$ минусов, то среди параметров столбцов будет ровно $k$ "-1". Ввиду того, что $k \leq [\frac{n}{2}]$, то харакеристика такой таблицы равна $k$. Значит характеристика таблицы $2 \times n$ принимает значения от 0 до $[\frac{n}{2}]$.

**Теорема 2.1.** $W_{2 \times n} = [0; \lfloor \frac{n}{2} \rfloor]$

## 3 Алгоритм нахождения характеристики заданной таблицы $m \times n$ с помощью производной таблицы

### 3.1 Производные таблицы

Напомним, что *классом таблиц* мы называем множество, содержащее в себе все таблицы, каждая из которых может быть получена из любой другой таблицы в этом множестве путем применения операции смены знаков в произвольных рядах один или несколько раз. Тогда нахождение характеристики таблицы равосильно нахождению таблицы с минимальным количеством минусов в классе заданной таблицы.

*Правильной таблицей* будем называть таблицу, в которой количество минусов в каждом столбце $x_i \leq \lfloor \frac{n}{2} \rfloor$. Ясно, что любую неправильную таблицу можно привести к правильному виду путем смены знаков в столбцах: если в каком-то столбце $x_i > \lfloor \frac{n}{2} \rfloor$, то достаточно произвести смену знаков в нем и получить $n - x_i < \rfloor \frac{n}{2} \lfloor$ минусов.

Разобьем таблицу $m \times n$ на $m-1$ таблиц $2 \times n$. Назовем их $D_1, D_2, \dots D_{m-1}$. Таблица $D_1$ состоит из 1-й и 2-й строки исходной таблицы, таблица $D_2$ состоит из 2-й и 3-й строки исходной таблицы. И так далее: таблица $D_i$ состоит из $i$-й и $i+1$-й строки исходной таблицы. Определим *производную таблицу*, как таблицу, состоящую из параметров столбцов таблиц $D_1, D_2, \dots, D_{m-1}$ (*Рисунок 3*). Каждая $i$-я строка содержит параметры столбцов таблицы $D_i$. Ясно, что производная таблица будет иметь размеры $m - 1 \times n$, где $m - 1$ - количество строк, а $n$ - количество столбцов.

Если таблица $B$ является произвдной для таблицы $A$, таблицу $A$ будем называть исходной для таблицы $B$.

| | | | | | |
|---|---|---|---|---|---|
| + | + | - | + | - | + |
| + | - | + | - | - | + |
| + | - | + | + | + | - |
| + | - | - | - | + | + |

| | | | | | |
|---|---|---|---|---|---|
| + | - | - | - | + | + |
| + | + | + | - | - | - |
| + | + | - | - | + | - |

Рис. 3: Слева - таблица $4 \times 6$, справа - соответвтующая ей производная таблица

Узнаем, что происходит с производной таблицей при смене знаков в рядах исходной таблицы:

- При смене знаков в столбцах исходной таблицы, происходит также смена знаков в столбцах таблиц $D_i$. Однако по Лемме 1 их параметры постоянны, ввиду того что длина столбца равна 2 - четное число, следовательно при смене знаков в столбцах исходной таблицы, производная таблица не изменяется.
- При смене знаков в строках исходной таблицы, меняются параметры столбцов всех таблиц $D_i$, которые содержат в себе эту строку (согласно Лемме 2). Строку 1 и строку $n$ содержат только таблицы $D_1$ и $D_{n-1}$ соответственно. Тогда при смене знаков в строках 1 или $n$, изменяются на противоположные параметры таблиц $D_1$ или $D_{n-1}$ соответственно. Строки под номером $i : 2 \leq i \leq n - 1$ содержат сразу две таблицы - $D_{i-1}$ и $D_i$. Поэтому при смене знаков в этих строках, меняются параметры сразу двух таблиц - $D_i i - 1$ и $D_i$. В производной же таблице происходит смена знаков в строках, номера которых соооветствуют номерам таблиц, в которых изменяются параметры столбцов.

Таким образом, при любой смене знаков в исходной таблице, в производной таблице может производиться *либо смена знаков в первой и последней строчке, либо смена знаков в любых двух соседних строках.*

Следовательно для таблиц одного класса существует не более, чем $2^m$ значений произодной таблицы, где m - количество строк (так как каждая строка имеет либо изначальный вид, либо в ней изменены знаки, количество считается по правилу комбинаторного умножения $2 \cdot 2 \cdot \dots \cdot 2 = 2^m$). Покажем, что, изменяя соседние пары строк или крайние строки, можно получить все $2^m$ значений производной таблицы.

**Лемма 3.** *Пусть дано упорядоченное множество, состоящее из m знаков + и - (например* $\{+-++-++-\}$*). Тогда, путем изменения любой соседней пары знаков или крайних знаков на противоположные, можно получить все* $2^m$ *возможных комбинаций знаков.*

*Доказательство.* Достаточно показать, что с помощью заданных операций можно изменить один знак в множестве на противоположный. Пусть заданное множество имеет вид $\{a_1, a_2, \ldots, a_n\}$, где $a_i = 1$, если на этом месте в множестве стоит "+", и $a_i = -1$, если стоит "-".

Изменить знак первого и последнего элемента можно напрямую (по условию леммы).

Чтобы изменить знак числа $a_i, i \neq 1, i \neq n$, достаточно сначала изменить пару $(a_i, a_{i+1})$, потом пару $(a_{i+1}, a_{i+2})$ и так далее до последней пары $(a_{m-1}, a_m)$ (конечно, можно "шагать" и в сторону первого элемента, если это более рационально). После этих действий множество будет иметь вид $\{a_1, \ldots, -a_i, \ldots, a_{m-1}, -a_m\}]$. Осталось лишь изменить напрямую знак числа $a_m$, и мы получим множество, в котором изменен лишь знак числа $a_i$ относительно изначальной конфигурации.

□

Отсюда следует, что для таблиц одного класса существует ровно $2^m$ значений производной таблицы.

## 3.2 Восстановление количества минусов в исходной таблице по столбцу производной таблицы

Заметим следующее утверждение, которое напрямую следует из определения параметра столбца.

**Утверждение 1.** *Параметр столбца таблицы* $D_i$ *равен -1 тогда и только тогда, когда в этом столбце стоят числа разных знаков, и равен 1 тогда и только тогда, когда стоят числа одного знака.*

Утверждение справедливо, так как таблица $D_i$ имеет столбцы размера 2, и произведение их чисел отрицательно, если в столбце числа разных знаков, и положительно, если в столбце числа одного знака.

Покажем, как восстановить столбец исходной таблицы из производной таблицы. Обозначим числа, стоящие в столбце исходной таблицы за $a_1, a_2, \ldots, a_m$, а числа, стоящие в столбце производной таблицы за $d_1, d_2, \ldots, d_{m-1}$. Заметим, что если $d_i = 1$, то $a_i = a_{i+1}$, а если $d_i = -1$, то $a_i = -a_{i+1}$.

Зафиксируем $a_1 = x$. Тогда начиная с 1 для каждого числа $i$ проверяем: если $d_i = 1$, тогда числу $a_{i+1}$ присваиваем значение $a_i$, иначе присваиваем ему значение $-a_i$. Так, мы последовательно выражаем каждый следующий знак столбца через предыдущий, считывая производный столбец. Таким образом для каждого значения $x$ мы однозначно восстанавливаем столбец исходной таблицы. По условию $x$ принимает только 2 значения: 1 и -1. Значит каждому производному столбцу соответствует два значения количеств минусов в исходном столбце. Причем ясно, что два этих количества можно получить друг из друга с помощью смены знаков, т.е если первое количество $q$, тогда второе - $m - q$.

Чтобы найти минимальное количество минусов, которое соответствует такой производной таблице, необходимо восстановить оба количества минусов, которые соответствуют каждому производному столбцу, для каждого из них выбрать минимальное, и проссумировать их, получая количество минусов в таблице.

Заметив, что в восстановленной таблице количества минусов в столбцах каждое меньше, чем $[\frac{n}{2}]$ (по нашему построению), получим следующее утверждение.

**Утверждение 2.** *По производной таблице можно однозначно восстановить исходную правильную таблицу.*

Отсюда следует, что в одном классе различных правильных таблиц ровно $2^m$.

### 3.3 Алгоритм нахождения характеристики

В пункте 3.1 мы показали, что весь класс, содержащий заданную таблицу сводится к ровно $2^m$ правильных таблиц и к такому же количеству значений производных таблиц. Далее в пункте 3.2 мы показали, как найти минимальное количество минусов для каждого значения производной таблицы. Чтобы найти характеристику, нужно выбрать минимальное из найденных количеств.

Пусть задана таблица $m \times n$. Для того, чтобы найти ее характеристику, необходимо произвести следующие шаги:

1. Построить производную таблицу для этой таблицы
2. Перебрать все возможные производные таблицы, которые можно получить путем смены знаков в исходной таблице ($2^m$ значений - по два значения для каждой строки).
3. Для каждой возможной производной таблицы произвести следущие действия.
   - Для каждого столбца производной таблицы восстановить минимальное из двух возможных количеств минусов в столбце исходной табилцы.
   - Просуммировать все полученные количества минусов в столбцах, чтобы получить минимальное количество минусов во всей таблице.
4. Среди полученных количеств минусов, восстановленных по производной таблице, найти минимальное. Оно и будет являться характеристикой.

Реализацию этого алгоритма на C++ см. в **Приложении** или по ссылке

`https://github.com/artemious3/characteristics_of_tables`.

## 4 Все возможные характеристики таблицы $3 \times n$

### 4.1 Производные и правльные таблицы $3 \times n$

**Определение.** *Таблицу $3 \times n$ будем называть* ***правильной****, если в любом ее столбце либо 0, либо 1 минусов.*

| | | | | |
|---|---|---|---|---|
| + | + | - | + | - |
| - | - | - | + | + |
| + | - | - | + | - |

| | | | | |
|---|---|---|---|---|
| + | - | + | + | + |
| - | + | + | + | - |
| + | + | + | + | + |

Рис. 4: Пример: неправильная и соответствующая ей правильная таблица

Далее мы будем рассматривать только правильные таблицы.

Разобьем таблицу $3 \times n$ на две таблицы $2 \times n$ - *верхнюю и нижнюю* таблицы. Верхняя таблица образована 1-й и 2-й строкой исходной таблицы. Нижняя таблица образована 2-й и 3-й строкой таблицы(*Рис. 5*).

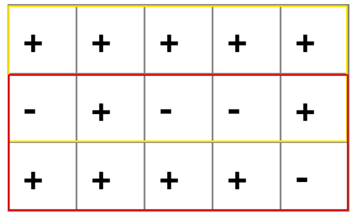

Рис. 5: Верхняя таблица обозначена желтым, нижняя – красным

Тогда в первой строке производной таблицы для этой таблицы будут находиться параметры столбцов верхней таблицы, во второй – параметры столбцов нижней таблицы.

| | | | |
|---|---|---|---|
| $\varepsilon_{1:x,1}$ | $\varepsilon_{1:x,2}$ | $\cdots$ | $\varepsilon_{1:x,n}$ |
| $\varepsilon_{2:x,1}$ | $\varepsilon_{2:x,2}$ | $\cdots$ | $\varepsilon_{2:x,n}$ |

Здесь $\varepsilon_{1:x,i}$ – параметр столбца $i$ верхней таблицы, а $\varepsilon_{2:x,i}$ – соответственно нижней.

## 4.2 Установление однозначного взаимного соответствия между правильной таблицей $3 \times n$ и производной таблицей

Рассмотрим все возможные варианты расстановки минусов в столбце производной таблицы. Восста-

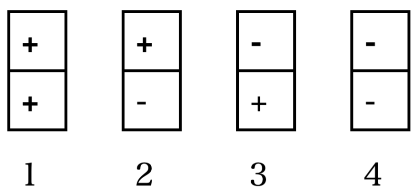


Рис. 6: Все возможные расположения знаков в столбцах таблицы $3 \times n$

новим количество "-1" в таких столбцах по алгоритму из пункта 3.2. Пусть числа в рассматриваемом столбце исходной таблицы – $a_1, a_2, a_3$, а количество "$-1$" в нём – $x$.

- Случай 1. Пусть $a_1 = 1$. Тогда по производному столбцу $a_2 = a_3 = 1$. Следовательно количество минусов $x = 0$ - подходит под определение правильной таблицы. Тогда случаю 1 соответсвует количество минусов $x = 0$.

- Случай 2. Пусть $a_1 = 1$. Тогда по производному столбцу $a_2 = 1; a_3 = -1$. Следовательно количество минусов $x = 1$. - подходит под определение правильной таблицы. Тогда этому случаю соответсвует количество минусов $x = 1$.

- Случай 3. Пусть $a_1 = -1$. Тогда по производному столбцу $a_2 = 1; a_3 = 1$. Следовательно количество минусов $x = 1$ - подходит под определение правильной таблицы. Тогда этому случаю соответсвует количество минусов $x = 1$.

- Случай 3. Пусть $a_1 = 1$. Тогда по производному столбцу $a_2 = -1; a_3 = -1$. Следовательно количество минусов $x = 1$ - подходит под определение правильной таблицы. Тогда этому случаю соответсвует количество минусов $x = 1$.

Случаи пронумерованы и представлены на Рисунке 6.

Отметим, какие столбцы правильной таблицы соответствуют приведенным случаям.

- Случаю 1 соответсвует столбец только с "1"
- Случаю 2 соответсвует столбец с одной "-1" в нижней клетке
- Случаю 3 соответсвует столбец с одной "-1" в верхней клетке
- Случаю 4 соответсвует столбец с одной "-1" в средней клетке

## 4.3 Установление соответствия между сменой знаков в рядах исходной таблицы и сменой знаков в рядах производной таблицы

Так как строки производной таблицы имеют всего по два варианта расположения чисел в них (в каждой строке может быть произведена лишь смена знаков), то всего различных вариантов расположения чисел в производной таблице $2 \cdot 2 = 4$. Приведем эти варианты ниже:

| | | | |
|---|---|---|---|
| $\varepsilon_{1:x,1}$ | $\varepsilon_{1:x,2}$ | $\cdots$ | $\varepsilon_{1:x,n}$ |
| $\varepsilon_{2:x,1}$ | $\varepsilon_{2:x,2}$ | $\cdots$ | $\varepsilon_{2:x,n}$ |

| | | | |
|---|---|---|---|
| $-\varepsilon_{1:x,1}$ | $-\varepsilon_{1:x,2}$ | $\cdots$ | $-\varepsilon_{1:x,n}$ |
| $\varepsilon_{2:x,1}$ | $\varepsilon_{2:x,2}$ | $\cdots$ | $\varepsilon_{2:x,n}$ |

| | | | |
|---|---|---|---|
| $\varepsilon_{1:x,1}$ | $\varepsilon_{1:x,2}$ | $\cdots$ | $\varepsilon_{1:x,n}$ |
| $-\varepsilon_{2:x,1}$ | $-\varepsilon_{2:x,2}$ | $\cdots$ | $-\varepsilon_{2:x,n}$ |

| | | | |
|---|---|---|---|
| $-\varepsilon_{1:x,1}$ | $-\varepsilon_{1:x,2}$ | $\cdots$ | $-\varepsilon_{1:x,n}$ |
| $-\varepsilon_{2:x,1}$ | $-\varepsilon_{2:x,2}$ | $\cdots$ | $-\varepsilon_{2:x,n}$ |

## 4.4 Подсчет количества непустых столбцов в производной таблице

По рассмотренным случаям, заметим, что минус в столбце исходной таблицы есть только тогда, когда в столбце производной табилце есть "-1";а также минусы в столбце исходной таблицы отсутствуют только тогда, когда в столбце производной таблицы отсутствуют "-1". Столбец, в котором отсутствуют "-1" будем называть пустым. Отсюда, количество минусов в правильной табилце $3 \times n$ равно количеству непустых столбцов в производной таблице.

Пусть задана правильная таблица $3 \times n$. Рассмотрим все 4 значения, которое принимает количество непустых столбцов в производных таблицах для таблиц из заданного класса.

1. Первое количество минусов в исходной таблице равно непосредственно изначальному количеству непустых столбцов в производной для нее таблицы, без изменения каких-либо строк. Из закономерностей, найденных в пункте II, в производной таблице количество столбцов с одной "-1" вверху равно $y_1$, с двумя "-1" – $y_2$, с одной "-1" внизу – $y_3$ (напомним, что $y_i$ - количесвто минусов в строке ). Все же остальные столбцы производной таблицы будут пустыми. Тогда изначальное количество непустых столбцов равно $y_1 + y_2 + y_3$.

2. Второе количество минусов равно количеству непустых столбцов в производной таблице, в которой произведена замена знаков в первой строчке. Заметим, что при замене знаков в верхней строчке производной таблицы, столбец $i$ становится пустым тогда и только тогда, когда в столбце $i$ в верхней клетке до замены стояла $-1$ (*Рисунок 7*, для удобства вместо самих чисел 1 или

-1, оставим только знак числа). Значит количество пустых столбцов после замены равно количеству столбцов, в которых есть лишь одна "-1" в верхней строчке до замены. Это количество непосредственно равно количеству "-1" в первой строчке исходной таблицы, т.е равно $y_1$. Непустыми будут все остальные столбцы. Их количество после замены знаков в данном случае будет равно $l_y - y_1$ (справедливо, так как количество столбцов производной таблицы равно количество столбцов исходной).

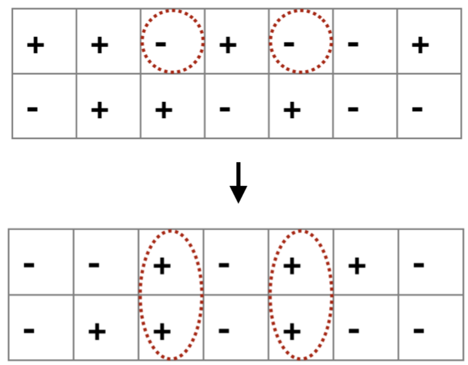

Рис. 7: При смене знаков в верхней строчке, столбец становится пустым, если в нем вверху был минус

3. Третье количество минусов равно количеству непустых столбцов в производной таблице, в которой произведена замена знаков во второй строчке. Тогда в производной таблице произойдет замена знаков в нижней строчке. Тогда количество непустых столбцов после замены знаков в данном случае считается аналогично предыдущему, и будет равно $l_y - y_3$ *(Рисунок 8)*.

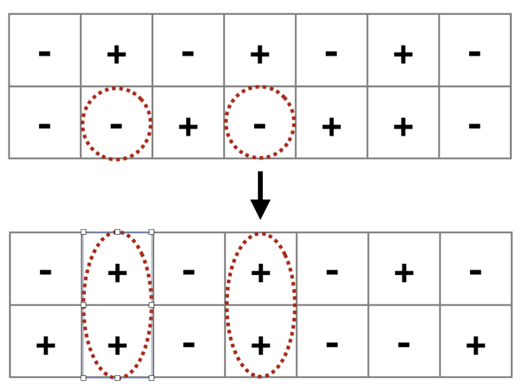

Рис. 8: При смене знаков в нижней строчке, столбец становится пустым, если в нем внизу был минус

4. Четвертое количество минусов равно количеству непустых столбцов в производной таблице, в которой произведена замена знаков в обоих строчках. Подсчитаем количество пустых столбцов после замены всех знаков в таблице. Заметим, что после замены всех знаков столбец становится пустым тогда и только тогда, когда до замены в этом столбце находилось две "-1" *(Рисунок 9)*. Количество таких столбцов непосредственно равно количеству "-1" во второй строчке исходной таблицы, т.е равно $y_2$. Значит количество непустых столбцов в этом случае равно $l_y - y_2$.

Таким образом, количество минусов в правильной таблице принимает следующие значения:

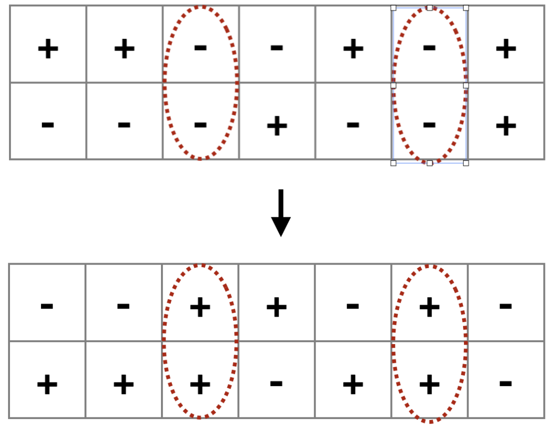

Рис. 9: При смене знаков в обоих строчках, столбец становится пустым, если в нем были все минусы

- $y_1 + y_2 + y_3$
- $l_y - y_1$
- $l_y - y_2$
- $l_y - y_3$

Характеристика таблицы $3 \times n$ равна

$$w = \min(y_1 + y_2 + y_3, l_y - y_1, l_y - y_2, l_y - y_3) \tag{1}$$

Таким образом, для Пункта 7 была доказана теорема, определяющая характеристику заданной таблицы $3 \times n$.

**Теорема 4.1.** *Пусть для правильной таблицы $3 \times n$, $y_i$ - количество минусов в строке $i$. Тогда для этой таблицы характеристика $w = \min(y_1 + y_2 + y_3, n - y_1, n - y_2, n - y_3)$.*

### 4.5 Все возможные характеристики для таблицы $3 \times n$

**Лемма 4.** *Для чисел $a_1, a_2, \dots, a_n$ таких, что $a_1 + a_2 + \dots + a_n = A$, верно $\max(a_1, a_2, \dots, a_n) \geq \frac{A}{n}$, причем равенство достигается тогда и только тогда, когда $a_1 = a_2 = \dots = a_n = \frac{A}{n}$*

*Доказательство.* Предположим противное: максимальное число $a_i$ меньше, чем $\frac{A}{n}$. Тогда все остальные числа $a_j \leq a_i < \frac{A}{n}$. Тогда в сумме $a_1 + a_2 + \dots a_i + \dots + a_n < n \cdot \frac{A}{n} = A$ равенство не достигается - противоречие. Значит максимальное число $a_i \geq \frac{A}{n}$ Теперь покажем, что равевенство достигается только тогда, когда $a_1 = a_2 = \dots = a_n = \frac{A}{n}$. Предположим противное: $\max(a_1, a_2, \dots, a_4) = \frac{A}{n}$, и есть какое-то $a_i \neq \frac{A}{n}$. Тогда в сумме $a_1 + \dots + a_i + \dots + a_n < n \cdot \frac{A}{n}$ равенство не достигается. □

По *Теореме 4.1* характеристика заданной таблицы $3 \times n$ равна $w = min(y_1 + y_2 + y_3, n - y_1, n - y_2, n - y_3)$. Пусть $a = n - (y_1 + y_2 + y_3)$. Тогда $w = \min(n - a, n - y_1, n - y_2, n - y_3) = n - max(a, y_1, y_2, y_3)$. Заметим, что $a + y_1 + y_2 + y_3 = n - (y_1 + y_2 + y_3) + y_1 + y_2 + y_3 = n$. Поэтому по *Лемме 4* $max(a, y_1, y_2, y_3) \geq \frac{n}{4}$. Поэтому $w = n - max(a, y_1, y_2, y_3) \leq n - \frac{n}{4} = \frac{3}{4}n$

**Утверждение 3.** *Для характеристики таблицы $3 \times n$ справедлива оценка $w \leq \frac{3}{4}n$*

Осталось показать, что для таблицы $3 \times n$ существуют все характеристики от 1 до $\lfloor \frac{3}{4}n \rfloor$.

**Лемма 5.** *Для чисел $a, b, c > 0$, $c \in \mathbb{Z}$ таких, что $a + b = c$ справедливо равенство $\lfloor a \rfloor + \lceil b \rceil = c$*

*Доказательство.* Для целых $a$ и $b$ равенство очевидно. Пусть теперь $a$ и $b$ нецелые. $\lfloor a \rfloor + \lceil b \rceil = [a] + [b] + 1 = a - \{a\} + b - \{b\} + 1 = a + b + 1 - (\{a\} + \{b\})$. Заметим, что так как в суммме $c = a + b = [a] + [b] + \{a\} + \{b\}$ число $[a] + [b]$ целое, то и число $\{a\} + \{b\}$ целое. Но так как числа $a$ и $b$ нецелые, единственное целое значение которое принимает число $\{a\} + \{b\} = 1$. Поэтому $a + b + 1 - (\{a\} + \{b\}) = a + b = c$ □

Построим таблицу $3 \times n$, характеристика которой равна $\lfloor \frac{3}{4}n \rfloor$. Для этого возьмем $y_1 = \lceil \frac{n}{4} \rceil$, $y_2 = \lceil \frac{n}{4} \rceil$, а $y_3 = \lfloor \frac{3}{4}n \rfloor - 2\lceil \frac{n}{4} \rceil$. Распишем значения, которые принимает количество минусов в такой таблице.

- $y_1 + y_2 + y_3 = \lfloor \frac{3}{4}n \rfloor$
- $n - y_1 = n - \lceil \frac{n}{4} \rceil$. По лемме 5 $\lfloor \frac{3}{4}n \rfloor + \lceil \frac{n}{4} \rceil = n$. Поэтому $n - y_1 = n - \lceil \frac{n}{4} \rceil = \lfloor \frac{3}{4}n \rfloor$
- Аналогично $n - y_2 = n - \lceil \frac{n}{4} \rceil = \lfloor \frac{3}{4}n \rfloor$.
- $n - y_3 = n - \lfloor \frac{3}{4}n \rfloor + 2\lceil \frac{n}{4} \rceil = \lceil \frac{n}{4} \rceil + 2\lceil \frac{n}{4} \rceil = 3\lceil \frac{n}{4} \rceil \geq \frac{3}{4}n > \lfloor \frac{3}{4}n \rfloor$

Характеристика такой таблицы равна $w = min(y_1 + y_2 + y_3, n - y_1, n - y_2, n - y_3) = \lfloor \frac{3}{4}n \rfloor$

Теперь если уменьшить одно из значений $y_i$ (например $y_1$) на число $\alpha$, тогда $y_1 + y_2 + y_3$ уменьшится и станет равным $y_1 + y_2 + y_3 - \alpha$. А значение $n - y_1$ увеличится, став равным $n - y_1 + \alpha$. Остальные же значения $n - y_2, n - y_3$ не изменятся. Чтобы получить таким образом все возможные характеристики нужно последовательно уменьшать количества минусов в строке на 1, пока оно не станет равным нулю (после того, как оно стало нулевым, необходимо продолжить уменьшать другое количество минусов). Тогда среди значений $y_1 + y_2 + y_3$, $n - y_1, n - y_2, n - y_3$ значение $y_1 + y_2 + y_3$ будет каждый раз уменьшаться на 1, а значения $n - y_1, n - y_2, n - y_3$ будут либо оставаться неизменными, либо увеличиваться. В таком случае значение $y_1 + y_2 + y_3$ всегда будет оставаться минимальным и будет являться характеристикой таблицы.

Таким образом в пункте **VI** была доказана теорема о всех возможных характеристиках таблицы $3 \times n$.

**Теорема 4.2.** $W_{3 \times n} = [0; \lfloor \frac{3}{4}n \rfloor]$

# 5 Оценка диапазона возможных характеристик для таблиц произвольной размерности

## 5.1 Оценка характеристики для таблицы $2k \times n$

Рассмотрим произвольную таблицу $2k \times n$. По *Лемме 1*, параметры столбцов этой таблицы остаются постоянными при изменении знаков в них, а по *Лемме 2* при изменении знаков в какой-либо из строк, все параметры столбцов изменяются на проитвоположные. Значит, для этой таблицы множество $E$ также принимает только 2 значения:

$$E_1 = \{\varepsilon_{x,1}, \varepsilon_{x,2}, ..., \varepsilon_{x,n}\}$$

$$E_2 = \{-\varepsilon_{x,1}, -\varepsilon_{x,2}, ..., -\varepsilon_{x,n}\}$$

Заметим следующий факт.

**Утверждение 4.** *Если параметр ряда $\varepsilon_i = -1$, тогда в нем есть как минимум одна "-1"(т.е количество минусов в этом ряду $a_i \geq 1$).*

Доказывается это тем, что если параметр ряда $\varepsilon_i = -1$, то в нем не могут отсутствовать "-1". Значит количество минусов во всей таблице не меньше, чем количество "-1" в множестве E. Но количество минусов в множестве E принимат только два значения. Значит характеристика таблицы не меньше, чем наименьшее из этих двух значений.

$$W \geq \min(x_{E_1}, x_{E_2}) \tag{2}$$

**Утверждение 5.** *Характеристика таблицы, у которой есть ряд четной длины, не меньше чем количество "-1" среди параметров этих рядов.*

Рассмотрим несколько частных случаев. Пусть в таблице $2k \times n$ есть $m$ столбцов ($m \leq [\frac{n}{2}]$), в которых ровно одна "-1", а во всех остальных столбцах все числа - "1". Тогда $x_{E_1} = m$, $x_{E_2} = n - m$. Тогда для этой таблицы в *Неравенстве 2* достигается равенство: в таблице уже есть m минусов, и уменьшить их количество по неравенству невозможно. А так как мы определили $m \leq [\frac{n}{2}]$, то $m < n-m$, следовательно для такой таблицы ее характеристика $w = m$.

Из этого следует следующее утверждение.

**Утверждение 6.** *Для таблицы $2k \times n$, существуют характеристики от 0 до $[\frac{n}{2}]$ (однако это не все возможные характеристики для таких таблиц).*

## 5.2 Оценка характеристики для таблицы $n \times m$ с помощью оценки количества минусов в рядах

Заметим, что в для каждого столбца таблицы, если количество минусов в нем $x_i \geq \frac{m}{2}$, то при смене знаков в данном ряду это количество станет $x_i \leq \frac{m}{2}$. Тогда минимальное количество минусов в каждом столбце не меньше чем $[\frac{m}{2}]$. Всего столбцов в таблице $n$. Значит характеристика таблицы $w \leq [\frac{m}{2}]n$. Аналогичная оценка применима и относительно строк таблицы: $w \leq [\frac{n}{2}]m$. В данном случае интерес представляет более сильная оценка из двух приведенных.

**Утверждение 7.** *Для произвольной таблицы $m \times n$ справедливы следующие неравенства: $w \leq [\frac{m}{2}] \cdot n$, $w \leq [\frac{n}{2}] \cdot m$*

## 5.3 Оценка характеристики для таблицы $n \times m$ с помощью ранга матрицы

Далее таблицу будем рассматривать как матрицу, строки и столбцы которой составляют числа записанные в соответствующих строках и столбцах таблицы. Здесь строки и столбцы мы также будем называть рядами матрицы. Пусть $A = (a_1 \quad a_2 \quad \ldots a_n)$ - ряд матрицы. Тогда произведением ряда $A$ на число $c$ будем называть ряд $A \cdot c = (ca_1 \quad ca_2 \quad \ldots \quad ca_n)$.
Определим сумму двух рядов одной длины (пусть длина равна $m$) следующим образом: $A + B = (a_1 + b_1, a_2 + b_2, \ldots, a_m + b_m)$. Аналогично вводится сумма трех и более рядов одной длины.
Будем говорить, что строки $(A_1, A_2, \ldots, A_n)$ составляют *систему строк матрицы $M$*, а множество $(B_1, B_2, \ldots, B_k)$ столбцов является *системой столбцов матрицы $M$*.
Будем говорить, что система $A$ строк $(A_1, A_2, \ldots, A_n)$ является *линейно независимой* если и только если выражение $\alpha_1 A_1 + \alpha_2 A_2 + \ldots \alpha_n A_n = 0$ (здесь в качестве 0 выступает строка, все числа которой равны 0) выполняется только на нулевом наборе $\alpha_i$-ых. Иначе, такую систему будем называть *линейно зависимой.*
*Базисом системы* строк будем называть подсистему исходной системы, такую что

1. Эта подсистема является линейно независимой.

2. Добавление любой строки в эту подсистему строк будет образовывать линейно зависимую систему.

Рангом системы строк $A$ будем называть количество строк составляющих базис системы строк($\operatorname{rank} A$). Аналогично введём понятие ранга системы столбцов ($\operatorname{rank} B$).
Определим *ранг матрицы* $\operatorname{rank} M$ как ранг системы ее строк. Из теоремы о ранге матрицы[1], имеем что $\operatorname{rank} A = \operatorname{rank} B$. Поэтому $\operatorname{rank} M = \operatorname{rank} A = \operatorname{rank} B$. Для каждой матрицы количество строк в любом базисе одинаково[1], поэтому и ранг для каждой матрицы определен однозначно.

**Лемма 6.** *Операции, определённые в задаче, не меняют ранга матрицы.*

*Доказательство.* Достаточно показать, что не изменится ранг системы строк. Не нарушая общности, скажем что базис системы строк составляет подсистема $\widetilde{A} = (A_1, A_2, \dots, A_q)$. Тогда уравнение $\alpha_1 A_1 + \alpha_2 A_2 + \dots + \alpha_i A_i + \dots \alpha_q A_q = 0$ имеет место только на нулевом наборе $\alpha_i$. Умножим строку $A_i$ на $-1$. Тогда ясно, что и уравнение $\alpha_1 A_1 + \alpha_2 A_2 + \dots - \alpha_i A_i + \dots \alpha_q A_q = 0$ тоже имеет место только на нулевом наборе $\alpha_i$. Условие 1) независимости систем выполнено. Возьмём строку $A_p, A_p \notin \widetilde{A}$. Тогда из условия 2) для системы $\widetilde{A}$ уравнение $\alpha_1 A_1 + \alpha_2 A_2 + \dots + \alpha_i A_i + \dots \alpha_q A_q = A_p$ имеет место на наборе $\widetilde{\alpha_i}$, таком что $\widetilde{\alpha_1}^2 + \widetilde{\alpha_2}^2 + \dots + \widetilde{\alpha_q}^2 > 0$ (т.е среди них есть хотя бы одно $\alpha_k \neq 0$). Тогда при умножении строки $A_i$ на $-1$ положим в предыдущем равенстве $\alpha_i = -\alpha_i$. Ясно, что тогда равенство всё равно будет выполнено и для строки $-1 \cdot A_i$ как и условие $\alpha_1^2 + \alpha_2^2 + \dots + \alpha_q^2 > 0$. Таким образом выполнено и условие 2) значит новая система тоже составляет базис, а следовательно ранг матрицы не изменился. □

**Теорема 5.1.** *Для любых размерностей $n \times k$ существует таблица с любой характеристикой от 0 до $min(n, k) - 1$.*

*Доказательство.* Рассмотрим таблицу

$$A = \begin{bmatrix} -1 & 1 & 1 & \dots & 1 & 1 & 1 & \dots & 1 \\ 1 & -1 & 1 & \dots & 1 & 1 & 1 & \dots & 1 \\ 1 & 1 & -1 & \dots & 1 & 1 & 1 & \dots & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1 & -1 & 1 & \dots & 1 \\ 1 & 1 & 1 & \dots & 1 & 1 & 1 & \dots & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1 & 1 & 1 & \dots & 1 \end{bmatrix} \qquad (1)$$

Обозначим количество $-1$ как $t$, и пусть обе размерности $A$ не меньше $t+1$ (в таком случае есть как минимум одна строка, состоящая из одних 1). Также не нарушая общности скажем, что количество строк в матрице не меньше количества столбцов. Покажем, что система $B$, состоящая из первых $t+1$ строк, составляет базис. Действительно, обозначим строку, в которой -1 расположена в столбце $i$ за $Q_i$. За $Q_0$ обозначим строку, следующую после строки $Q_t$(строка $Q_0$ состоит только из единиц). Тогда систему $B$ составляютс строки $Q_0, Q_1, Q_2, \dots, Q_t$.

Покажем, что равенство $\alpha_1 Q_1 + \alpha_2 Q_2 + \dots + \alpha_t Q_t + \alpha_{t+1} Q_0 = 0$ выполняется только на нулевом наборе $\alpha_i$-ых.

Для выполнения равенства необходимо выполнение системы уравнений;

$$\begin{cases} \alpha_2 + \dots + \alpha_t + \alpha_{t+1} = \alpha_1 \\ \alpha_1 + \alpha_3 + \dots + \alpha_t + \alpha_{t+1} = \alpha_2 \\ \dots \\ \alpha_1 + \dots + \alpha_{t-1} + \alpha_{t+1} = \alpha_t \\ \alpha_1 + \dots + \alpha_{t-1} + \alpha_t = -\alpha_{t+1} \end{cases}$$

Попарно отняв от первого уравнения каждое уравнение системы, получим.

$$\begin{cases} \alpha_2 - \alpha_1 = \alpha_1 - \alpha_2 \\ \alpha_3 - \alpha_1 = \alpha_1 - \alpha_3 \\ \dots \\ \alpha_t - \alpha_1 = \alpha_1 - \alpha_t \\ \alpha_{t+1} - \alpha_1 = \alpha_1 + \alpha_{t+1} \end{cases}$$

Откуда получаем $\alpha_1 = \alpha_2 = \dots = \alpha_t = \alpha$, а из последнего получаем $\alpha = 0$. Следовательно равенство $\alpha_1 Q_1 + \alpha_2 Q_2 + \dots + \alpha_t Q_t + \alpha_{t+1} Q_0 = 0$ выполняется только на нулевом наборе $\alpha_i$-ых.

Теперь покажем, что добавление любой строки матрицы A в нашу выбранную систему B сделает ее линейно зависимой. По нашему построению в матрице A остались лишь строки, состоящие только из 1. Но в выбранной системе уже есть строка, состоящая только из 1, поэтому равенство $\alpha_1 Q_1 + \alpha_2 Q_2 + \dots + \alpha_t Q_t + \alpha_{t+1} Q_0 = Q_p$, $Q_p$ - строка из единиц, $p \neq 0$, выполняется для $\alpha_1 = \dots = \alpha_t = 0$ и $\alpha_{t+1} = 1$ Поэтому добавление в выбранную систему строки из A сделает её линейно зависимой. Таким образом первые $t+1$ строк действительно составляют базис и соответственно её ранг - $t+1$.

Покажем, что с помощью данных нам операций не получится уменьшить число минусов у такой таблицы. Для этого предположим противное. Рассмотрим такую таблицу, в которой мы получили количество $-1$ меньше $t$. Тогда по принципу Дирихле у нас будет минимум $n-t+1$ строк состоящих из одних единиц. Но тогда ясно, что ранг такой матрицы будет меньше $t+1$(т.к. будет минимум $n-t+1$ одинаковых строк, только одна из которых может входить в базис). Из *Леммы 6* получаем противоречие. Поэтому характеристика таблицы $w_A = t$.

Таблицу такого вида мы можем составить для любых t от 1 до $min(n,k)-1$, следовательно для таблицы $m \times n$ существуют характеристики от 1 до $min(n,k)-1$.

□

Нетрудно заметить, что если в таблице в каждом ряду либо ровно по одному минусу, либо все плюсы (при этом есть как минимум одна строка и один столбец с одними $+$), то ранг системы строк матрицы, соответствующей этой таблице, также равен количеству рядов с одним минусом (аналогично, как в последней теореме). Пример - рисунок 10.

| | | | | | |
|---|---|---|---|---|---|
| + | + | - | + | + | + |
| + | + | + | - | + | + |
| + | + | + | + | + | - |
| + | + | + | + | + | + |

Рис. 10: Характеристика такой таблицы равна 3

**Утверждение 8.** *Если в классе заданной таблицы B есть таблица A, переменой местами строк и столбцов в которой, можно получить таблицу вида (1), то характеристика таблицы B равна количеству минусов в таблице A.*

*Доказательство.* Заметим, что перемена местами двух строк не меняет местами ранг по их системе. Ясно, что ранг по строкам не изменится, т.к. независимо от перемены местами строк базис их системы останется тот же. И т.к. ранги по строкам и по столбцам равны, то ранг матрицы не изменится.

Получили, что у таблиц, полученных из (1) путём замены строк и столбцов местами характеристика равна количеству минусов в таблице. Откуда и следует утверждение. □

Из утверждения 8 имеем **способ нахождения характеристики таблицы в матричном виде**. Он заключается в том, что если удалось привести таблицу к виду из утверждения 8, то мы найдём её характеристику. Ясно, что таким способом нельзя найти характеристики для всех таблиц, однако он может быть очень полезным для больших таблиц, в которых с его помощью можно быстро найти характерисику.

## 5.4 Оценка максимальной характеристики таблицы $m \times n$ с помощью перебора программой

Для понимания направления дальнейшего исследования в данной задаче, нами была создана прогамма, которая находит максимальную характеристику среди харакеристик 3000 случайно заданных таблиц (см. **Приложение. Принцип действия, пункт 3**) Так как рассматриваемые таблицы имеют относительно небольшие размеры, то среди 3000 случайно заданных таблиц с крайне высокой вероятностью встретится максимально возможная характеристика.

С помощью полученных данных (см. **Приложение 2**), мы построили следующий график (Рисунок 11).

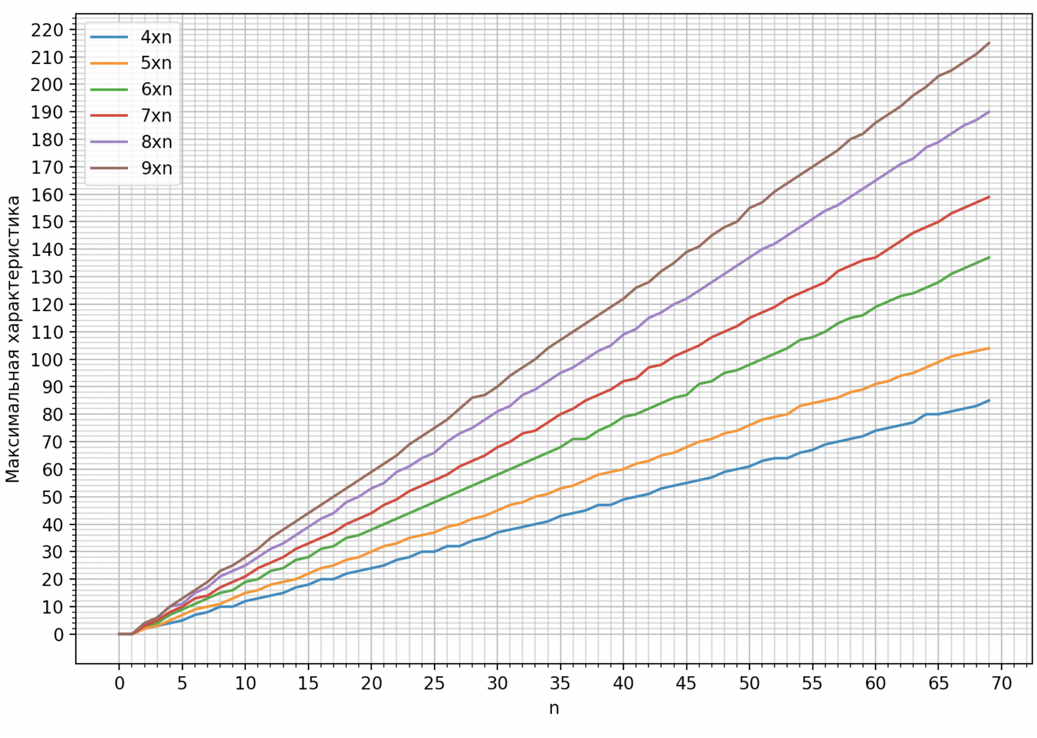


Рис. 11: Максимальные найденные характеристики для таблиц $i \times j; i \in [4; 10); j \in [1; 79)$

Также построен график отношения максимальной характеристики к произведению размерностей $m \cdot n$.

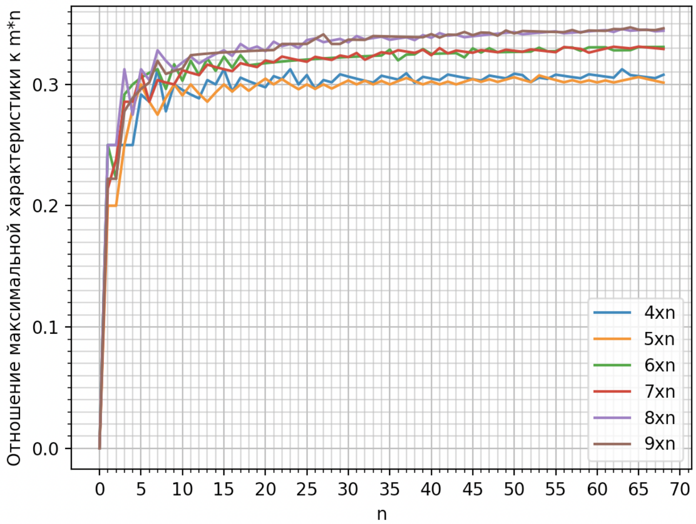


Рис. 12: Отношение максимальной характеристики к $m \times n$

На основании этого можно выдвинуть гипотезу о том, что для заданного $m$ при достаточно больших $n$ отношение максимальной характеристики к $m \cdot n$ постоянно. Это позволяет приблизительно оценить характеристику таблицы $m \times n$ для больших $n$, лишь один раз посчитав отношение $\frac{w_{\max}}{m \cdot n}$ для некоторого большого n. Например, из графика видно, что для таблицы $4 \times n$ и $5 \times n$ справедлива оценка

$$\frac{w_{\max}}{m \cdot n} \approx 0.3$$

Поэтому для таких таблиц

$$w_{\max} \approx 0.3mn$$

# 6 Выводы

- Диапазон возможных характеристик таблицы $2 \times n$ — $(1, 2, ..., \lfloor \frac{n}{2} \rfloor)$
- Диапазон возможных характеристик таблицы $3 \times n$ — $(1, 2, ..., \lfloor \frac{3}{4} n \rfloor)$
- Для таблицы $2k \times n$ среди возможных характеристик есть $(0{,}1, ..., \lfloor \frac{n}{2} \rfloor)$
- Для таблицы $m \times n$ среди возможных характеристик есть все от 0 до ее минимальной размерности минус 1, т.е $[0; \min(m, n) - 1] \in W_{m \times n}$
- Для характеристики любой таблицы $m \times n$ справедливы оценки $w \leq [\frac{m}{2}] \cdot n$ и $w \leq [\frac{n}{2}] \cdot m$
- Для правильной таблицы $3 \times n$ характеристика $w = \min(y_1 + y_2 + y_3, n - y_1, n - y_2, y - x_3)$, где $y_i$ - количество минусов в строке $i$.
- Для таблиц $4 \times n$ и $5 \times n$ максимальная характеристика $w \approx 0.3mn$

## Список литературы

[1] А.И.Кострикин. Введение в алгебру. Часть 1. Основы алгебры. *с.74, Теорема 1*, *с.71, Теорема 2.*

[2] `https://ru.wikipedia.org/wiki/Отношение_эквивалентности`

# Приложение

Данная прогрмамма находит характеристику заданной пользователем таблицы с помощью алгоритма, описанного выше в данной работе.

**Принцип действия:** В программе есть три функции, которые выбираются пользователем в начале работы программы

1. Программа случайным образом задает некоторое количество таблиц (это количество указывается в первом аргументе данной функции, по умолчанию - 1000), а затем находит для каждой из них характеристику, далее выводит на экран максимальную найденную характеристику.

2. Программа находит характеристику заданной пользователем таблицы. Сначала пользователю требуется ввести размеры задаваемой таблицы, после чего построчно вводить знаки таблицы. Когда пользователь ввел одну строку, программа "подчеркивает" ее в командной строке. Затем команда выводит характеристику заданной таблицы, а также выводит маску соответствующей ей производной таблицы (т.е в каких строках в производной таблицы необходимо произвести замену знаков, чтобы эта производная таблица соотвествовала таблице с минимальной характеристикой.

3. Программа запускает функцию 1 для каждой таблицы $i \times j$, где $i \in [2; m); j \in [4; n)$, и записывает полученные данные в файл.

**Алгоритм перебора всех значений производной таблицы**: Представим множество строк производной таблицы следующим образом:10010...0101.Такое представление назовем *маской*. Здесь i-е число равно 1,если в этой строке в производной таблице произведена смена знаков относительно изначальной таблицы, и равно 0 - если не произведена. Так как выше мы доказали, что определенными в задаче операциями можно получить любую комбинацию строк прозводной таблицы, то алгоритм можно свести к перебору всех двоичных чисел от 000...0 до 111...11, где количество цифр равно количеству строк. Вполне очевидно, что среди всех этих чисел встретится любая комбинация из m нулей и единиц. Для этого была написана функция nextValue, которая возвращает следующее значение таблицы на основе предыдущего.

**Технические моменты**:

1. Таблица хранится в виде двумерного массива bool, где значение true соответсвует знаку "-", а значение false – знаку "+".

2. Для удобного перебора всех возможных производных таблиц, производная таблица имеет длину строк, равную m+1. Это сделано для того, чтобы m плюс первое значение строки хранило информацию о том, произведена ли в этой строке смена знаков относительно изначальной производной таблицы: true - произведена, false - не произведена (см. Алгоритм перебора всех значений производной таблицы). То есть m+1 столбец производной таблицы содержит ее маску.

3. После получения характеристки таблицы, программа выводит маску таблицы с минимальной характеристикой и для удобства выводит также изначальную производную таблицу. С помощью изначальной производной таблицы и маски, можно понять, какие строки изменены в таблице с минимальным количеством минусов.

```
#include <iostream>
#include <cmath>
#include <cstdlib>
#include <ctime>
#include <fstream>

using namespace std;

bool** maxTable;
int macCh;

//Обыкновенный вывод таблицы в командную строку.
void tableOutput(bool** table, int n, int m)
{
    for(int i = 0; i < n; i++)
    {
        for(int j = 0; j < m; j++)
        {
            if(table[i][j])
            cout << "- ";
            else
            cout << "+ ";
        }
        cout << endl;
    }
}


bool* changeSigns(bool* raw, int m)
{
    bool* newraw = new bool[m+1];
    for(int i = 0; i < m+1; i++)
        newraw[i] = !raw[i];
    delete[] raw;
    return newraw;
}


//Функция аналогична прибавлению к маске 1.
//Она ищет в ней первое ненулевое число,
//делает его ненулевым, и делает нулевыми
//все предыдущие числа.
int nextValue(bool** table, int m, int n)
{
    for(int i = 0;i < n; i++)
    {
        if(table[i][m] == 0)
        {
            for(int j = i;j >= 0; j--)
            {
                table[j] = changeSigns(table[j], m);
```

```
            }
            return 0;
        }
    }
    return -1;
}

//Функция, которая возвращает производную таблицу
bool** makeDerTable(bool** table, int m, int n)
{
    bool** derTable = new bool*[n-1];
    for(int i = 0; i < n-1; i++)
    {
        derTable[i] = new bool[m+1];
        for(int j = 0; j < m; j++)
        {
            //В производной табилице стоит плюс (0), если соседние числа равны
            //и стоит - (1), если они не равны.
            derTable[i][j] = !(table[i][j] == table[i+1][j]);
        }
        derTable[i][m] = 0;
    }
    return derTable;
}

//Реализация алгоритма восстановления количества минусов в столбце
//из производного столбца
int getAmountFromDer(bool** dtable, int column, int n, int &nhalf)
{
    int c = 1;
    bool isCurrent = true;
    for(int i = 0; i < n-1; i++)
    {
        if(dtable[i][column] == 0 and isCurrent)
        c++;
        else if (dtable[i][column] == 1)
        {
            isCurrent = !isCurrent;
            if(isCurrent == true)
            c++;
        }
    }
    if(c > nhalf)
    c = n-c;

    return c;
}


int findCharacteristic(bool** table, int m, int n, int &valueCount, bool** derTable)
{
```

```
    //устанавливаем большое значение характеристики,
    // чтобы далее сравнивать его с гарантированно меньшими значеними*
    int charact = m*n;
    int nhalf = floor(n/2);
    int c = 0;
    //переменная valueCounter хранит, сколько раз к маске прибавили 1.
    int valueCounter;
    do
    {
        int amount = 0;
        for(int i = 0; i < m; i++)
        {
            amount += getAmountFromDer(derTable, i, n, nhalf);
        }
        if(amount < charact)
        {
            charact = amount;
            valueCounter = c;
            //Вместе с присвоением переменной charact минимальной
            //на данный момент характеристики, переменной valueCounter
            //присваивается значение маски
            //(ибо маска - двоичное число) для этой
            //минимальной характеристики,
            //чтобы потом по маске найти способ получить таблицу с
            //минимальной характеристкой из исходной
        }
        c++;
    } while (nextValue(derTable, m, n-1) != -1);
    valueCount = valueCounter-1;
    return charact;
}


bool swapValues(int &a, int &b)
{
    if(a > b)
    {
        int temp = b;
        b = a;
        a = temp;
        return true;
    }
    return false;
}

//Функция ввода таблицы через командную строку. (mr и nr - ссылки на переменные m и n вне этой функции)
bool** inputTable(int &mr, int &nr)
{
    int m, n;
    cout << "n (raws):"; cin >> n;
    cout << "m (columns):"; cin >> m;
```

```
    mr = m; nr = n;
    bool** table = new bool*[n];
    for(int i = 0; i < n; i++)
    {
        //m+1 столец таблицы -- ее маска (см. технические моменты)
        table[i] = new bool[m+1];
        for(int j = 0;j < m;j++)
        {
            char t;
            cin >> t;
            if(t == '-')
            table[i][j] = 1;
            else
            table[i][j] = 0;
        }
        table[i][m]=0;

        for(int i = 0; i < m; i++)
        cout << "__";
        cout << endl;
    }
    return table;
}

//Функция возвращает таблицу со случайно заданными знаками
bool** randomTable(int m, int n)
{
    bool** table = new bool*[n];
    for(int i = 0; i < n; i++)
    {
        table[i] = new bool[m+1];
        for(int j = 0;j < m;j++)
        {
            table[i][j] = rand() % 2;
        }
        table[i][m]=0;
    }
    return table;
}

int setTableProgram_()
{
    int m, n;
    int valueCounter;
    bool* mask;
    bool** table = inputTable(m, n);
    bool** derTable = makeDerTable(table, m, n);;
    cout << "CHARACHTERISTIC: " << findCharacteristic(table, m, n, valueCounter, derTable) << endl;
    for(int i = 0; i < n-1;i++)
    {
        derTable[i] = changeSigns(derTable[i], m);
```

```
    }
    cout  << "FOR DERRIVATIVE TABLE: \n";
    tableOutput(derTable, n-1, m);
    for(int i = 0; i <= valueCounter; i++)
    {
        nextValue(derTable, m, n-1);
    }
    cout << "WITH MASK:";
    for(int i = 0; i < n-1; i++)
    cout << derTable[i][m];
    cout << endl;
    return 0;
}

int randTableProgram_(int amountOfTables, bool isInput, int mr, int nr, int* gotChar)
{
    int m, n;
    bool** derTable;
    if(isInput)
    {
    cout << "n(raws):"; cin >> n;
    cout << "m(columns):"; cin >> m;
    }
    else
    {
        m = mr;
        n = nr;
    }

    int valueCounter;
    bool* mask;
    int maxChar = 0;
    for(int q = 0; q < amountOfTables; q++)
    {
        bool** table = randomTable(m, n);
        derTable = makeDerTable(table, m, n);
        int valueCounter;
        int c = findCharacteristic(table, m, n, valueCounter, derTable);
        //cout << q << endl;
        if(c > maxChar)
        {
            maxChar = c;
            maxTable = new bool*[n];
            for(int i = 0; i < n; i++)
            {
                maxTable[i] = new bool[m];
                for(int j = 0; j < m; j++)
                {
                    maxTable[i][j] = table[i][j];
                }
            }
```

```
        }

    }
    cout << "FOUND TABLE: \n";
    tableOutput(maxTable, n, m);
    if(isInput)
    {
        cout << "WITH CHARACTERISTIC:" << maxChar << endl;
    }
    else
    *gotChar = maxChar;
    return 0;
}


int goThroughTables_(int nmax, int mmax)
{
    ofstream my_file;
my_file.open("tabledata.txt");
    cout << "CREATED FILE" << endl;
        for(int i = 2; i < mmax; i++)
        {
            my_file << i;
            for(int j = 4; j < nmax; j++)
            {
                int* charr_ = new int;
                randTableProgram_(3000, false, i, j, charr_);
                int charr = *charr_;
                delete charr_;
                my_file << " " << charr;
                cout << "WENT TRHOUGH  " << i << "x" << j << " TABLES" << endl;
            }
            my_file << endl;
        }
    my_file.close();
    return 0;
}

int main()
{
    int option;
    cout << "1 GET MAX CHARACTERISTIC OF RANDOM TABLE \n";
    cout << "2  GET MAX CHARACTERISTIC OF INPUT TABLE \n";
    cout << "3 GET FILE WITH MAX CHARACTERISTIC FOR TABLES (from 2 to m)x(from 4 to n)\n";
    cin >> option;
    switch (option)
    {
    case 1:
        return randTableProgram_(1000, true, 0, 0, nullptr);

    case 2:
```

```
        return setTableProgram_();

    case 3:
    {
        int n, m;
        cout << "N:"; cin >> n;
        cout << "M:"; cin >> m;
        return goThroughTables_(n, m);
    }

    default:
        cout << "WRONG INPUT\n";
        break;
    }
    srand(time(NULL));

    //return goThroughTables_(10, 70);
    //return setTableProgram_();
}
```

## Приложение 2.

Файл с данными, по которым был построен график с Рисунка 11. В первом столбце расположены значения переменной $n$. В столбцах со 2 по 6 расположены максимальные найденные характеристики таблиц от $4 \times n$ до $9 \times n$ соответственно.

0 0 0 0 0 0 0
1 0 0 0 0 0 0
2 2 2 3 3 4 4
3 3 3 4 5 6 6
4 4 5 7 8 10 10
5 5 7 9 10 11 13
6 7 9 11 13 15 16
7 8 10 13 14 17 19
8 10 11 15 17 21 23
9 10 13 16 19 23 25
10 12 15 19 21 25 28
11 13 16 20 24 28 31
12 14 18 23 26 31 35
13 15 19 24 28 33 38
14 17 20 27 31 36 41
15 18 22 28 33 39 44
16 20 24 31 35 42 47
17 20 25 32 37 44 50
18 22 27 35 40 48 53
19 23 28 36 42 50 56
20 24 30 38 44 53 59
21 25 32 40 47 55 62
22 27 33 42 49 59 65
23 28 35 44 52 61 69
24 30 36 46 54 64 72
25 30 37 48 56 66 75
26 32 39 50 58 70 78
27 32 40 52 61 73 82
28 34 42 54 63 75 86
29 35 43 56 65 78 87
30 37 45 58 68 81 90
31 38 47 60 70 83 94
32 39 48 62 73 87 97
33 40 50 64 74 89 100
34 41 51 66 77 92 104
35 43 53 68 80 95 107
36 44 54 71 82 97 110
37 45 56 71 85 100 113
38 47 58 74 87 103 116
39 47 59 76 89 105 119
40 49 60 79 92 109 122
41 50 62 80 93 111 126
42 51 63 82 97 115 128
43 53 65 84 98 117 132
44 54 66 86 101 120 135
45 55 68 87 103 122 139
46 56 70 91 105 125 141
47 57 71 92 108 128 145
48 59 73 95 110 131 148
49 60 74 96 112 134 150
50 61 76 98 115 137 155
51 63 78 100 117 140 157
52 64 79 102 119 142 161
53 64 80 104 122 145 164
54 66 83 107 124 148 167
55 67 84 108 126 151 170
56 69 85 110 128 154 173
57 70 86 113 132 156 176
58 71 88 115 134 159 180
59 72 89 116 136 162 182
60 74 91 119 137 165 186
61 75 92 121 140 168 189
62 76 94 123 143 171 192
63 77 95 124 146 173 196
64 80 97 126 148 177 199
65 80 99 128 150 179 203
66 81 101 131 153 182 205
67 82 102 133 155 185 208
68 83 103 135 157 187 211
69 85 104 137 159 190 215