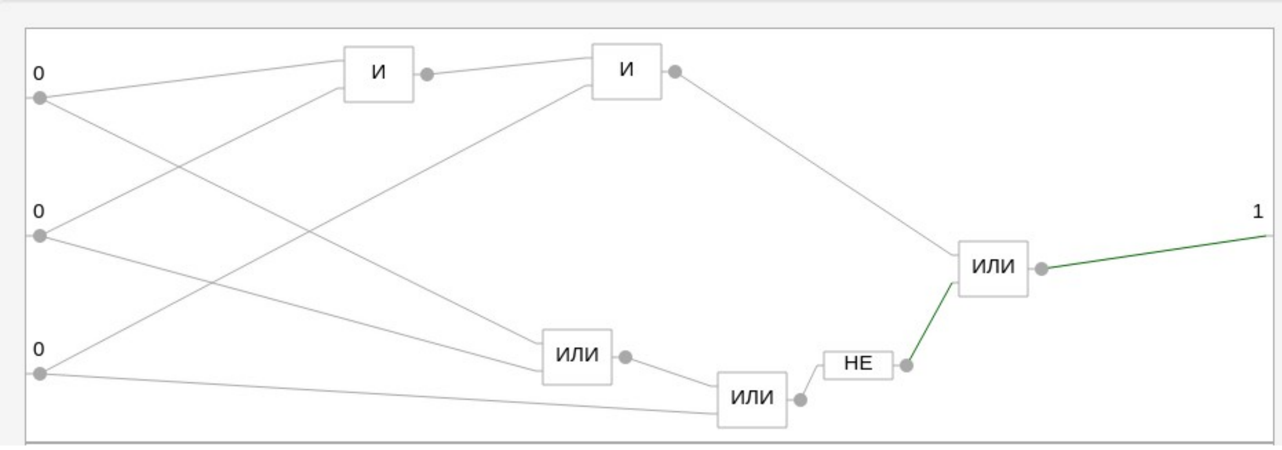
Олимпиада по Дискретной Математике
и Теоретической Информатике

Задача 1. (3 балла) При помощи логических элементов **И**, **ИЛИ** и **НЕ** постройте логическую схему, с тремя входами и одним выходом. На вход подаются нули или единицы; на выходе должна быть единица тогда и только тогда, когда значения входов совпадают.

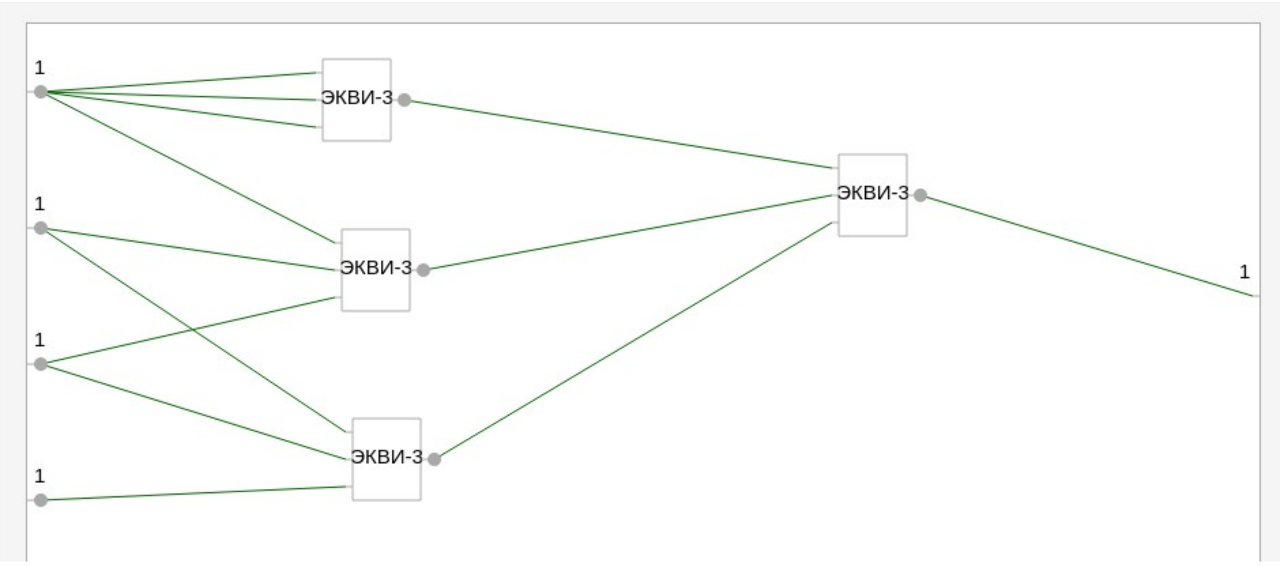
Решение:



Обозначим входы системы за x, y и z. (x И y И z) истинно когда все три входа истинны. НЕ(x ИЛИ y ИЛИ z) истинно когда все три элемента ложны.

Задача 2. **(3 балла)** Логический элемент ЭКВИ-3 работает так: на вход подаются нули или единицы; на выходе оказывается единица тогда и только тогда, когда значения входов совпадают.
С помощью логического элемента ЭКВИ-3 постройте логическую схему с четырьмя входами и одним выходом, реализующую элемент ЭКВИ-4: на выходе оказывается единица тогда и только тогда, когда значения всех четырёх входов совпадают

Решение:



Верхний элемент ЭКВИ-3 получает на вход три одинаковых сигнала и поэтому всегда истинный. Соответственно, для истинности всей схемы необходимо и достаточно, чтобы второй и третий элементы ЭКВИ-3 также принимали истинное значение, а это происходит в точности когда все входы схемы совпадают.

Задача 3. Логический элемент ЭКВИ-3 работает так: на вход подаются нули или единицы; на выходе оказывается единица тогда и только тогда, когда значения входов совпадают.
а) (3 балла) Докажите, что при помощи элемента ЭКВИ-3 можно построить схему с одним выходом и любым количеством входов, выдающую аналогичный результат: на выходе оказывается единица тогда и только тогда, когда значения всех входов совпадают
б) (4 балла) Докажите, что такую схему с *n* входами можно построить, использовав не более *n* элементов ЭКВИ-3.

Решение:
Будем строить схему аналогично решению предыдущей задачи. Обозначим значения входов за $x_1, x_2, \ldots, x_n$.
Построим элементы y_1=ЭКВИ(x_1, x_2, x_3), y_2=ЭКВИ(x_3, x_4, x_5), … ; предпоследний элемент имеет вид
ЭКВИ($x_{n-3}, x_{n-2}, x_{n-1}$) или ЭКВИ($x_{n-4}, x_{n-3}, x_{n-2}$) в зависимости от чётности *n*. Последний элемент в любом случае будет иметь вид ЭКВИ(x_{n-2}, x_{n-1}, x_n). Их количество в любом случае равно целой части n/2, которую мы обозначим за [n/2]. Введём ещё одно обозначение: N=n+1. Это обозначение кажется странным на первый взгляд, но оно упростит нам вычисления.
Итак, теперь у нас [(N-1)/2] элементов.

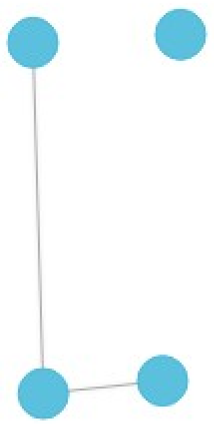
Кроме того, построим элемент y_0=ЭКВИ(x_1, x_1, x_1) - всегда истинный.
Теперь у нас [(N-1)/2]+1 элементов и нам надо убедиться, что они всегда истинны.
Аналогичным образом построим для них ещё [([(N-1)/2]+1-1) /2]=[(N-1)/4]. Добавим к ним уже построенный ранее истинный элемент и из этих [(N-1)/4]+1 элемента получим [(N-1)/8] новых. Будем продолжать так, пока на одном из шагов мы не получим
[(N-1)/2^k]=1 элемент. Общее количество построенных элементов составит 1 + [(N-1)/2] +[(N-1)/4]+...+[(N-1)/2^k], что не больше, чем та же самая сумма без знаков целой части, которая, в свою очередь, меньше 1+(N-1)=N. То есть, общее количество построенных элементов строго меньше N=n+1, то есть не больше n.

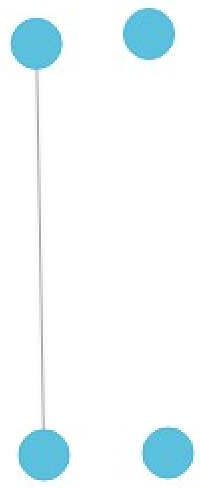
Задача 4. Графом называется рисунок, состоящий из точек (вершин), соединённых отрезками (рёбрами). При этом нам не важно, как именно нарисован граф, важно только, какие вершины с какими соединены.

а) (3 балла) Учитель нарисовал на доске граф. Три ученика перерисовали его в тетрадки (возможно, не сохранив расположение вершин), при этом каждый из них потерял по одной вершине (каждый — свою) и все выходящие из неё рёбра.
Оказалось, что по рисункам ребят нельзя восстановить нарисованный учителем граф.
Как такое могло быть? Постройте пример трёх рисунков учеников и двух возможных рисунков учителя.

б) (6 баллов) Верно ли, что для любого *n* можно придумать набор из *n* таких рисунков, по которым исходный граф восстанавливается неоднозначно (то есть существует два и более варианта)?

Решение:





а) Самое просто решение пункта а) изображено на рисунке.
Если из левого графа удалить вершину степени 2, получится граф, состоящий из трёх изолированных вершин. То же самое получается, если из правого графа удалить вершину степени один. Таким образом, три изолированных вершины — рисунок первого ученика.

Второй и третий рисунки одинаковы. Они получаются, если из левого графа удалить вершину степени 1, а из правого графа — изолированную.

б) Решение в общем случае получается многократным копированием этого.
Построим граф, состоящий из k копий правого графа и k-1 копии левого и второй граф, состоящий, наоборот, из k копий левого графа и k-1 копии правого.

Тогда при удалении из первого графа вершины степени 2 (которых в нём k штук) получается то же самое, что и при удалении из второго графа вершины степени 1 (которых там $2k$ штук). Это уже k совпадающих рисунков.

Кроме того, при удалении из первого графа вершины степени 1 (которых в нём $2k$ штук) получается то же самое, что и при удалении из второго графа вершины степени 0 (которых там тоже $2k$ штук). Это ещё $3k$ совпадающих рисунков.

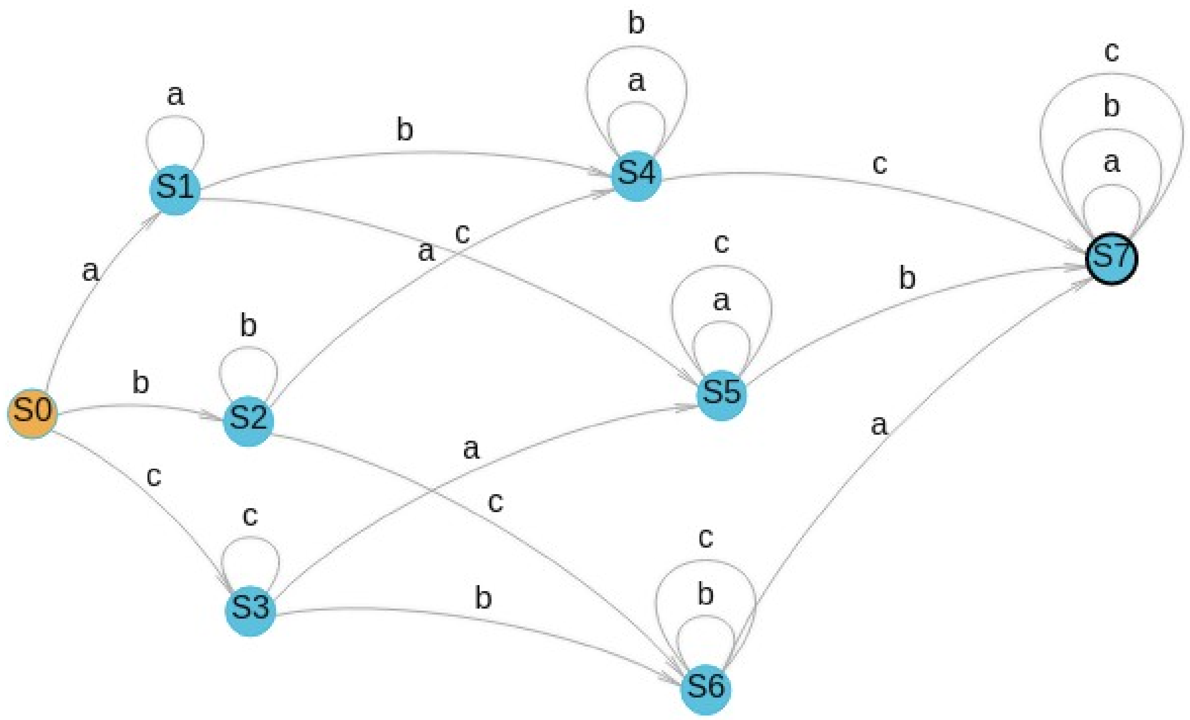
Таким образом, для построения нужного графа достаточно взять *k* хотя бы *n*/3.

Задача 5.(3 балла) Графом называется рисунок, состоящий из точек (вершин), соединённых отрезками (рёбрами). При этом нам не важно, как именно нарисован граф, важно только, какие вершины как соединены.
Подграф, который получается из графа удалением одной вершины, назовём ***картой***, а набор всех карт графа — ***колодой***.
В задаче отборочного тура мы уже доказывали, что по колоде можно восстановить общее количество рёбер графа. Также легко доказать, что можно установить ***степени*** всех вершин (количество выходящих из неё рёбер): степень вершины, которой не хватает на карте — это в точности количество отсутствующих на ней рёбер.
Докажите, что граф, степени всех вершин которого чётны, однозначно восстанавливается по своей колоде.

Решение: Как указано в условии задачи, мы можем установить степени всех вершин в графе. Соответственно, мы можем понять, что степени всех вершин чётны. После этого посмотрим на любую карту. Отсутствующая на ней вершина соединена в точности со всеми вершинами, у которых на этой карте нечётная степень.

Задача 6. (3 балла) Постройте конечный автомат, распознающий те и только те слова в алфавите **{a, b, c}**, которые содержат все три буквы алфавита.

Решение:



Состояния **S1**, **S2** и **S3** соответствуют словам, состоящих только из одной буквы: **a**, **b** или **c** соответственно. Состояния **S4**, **S5** и **S6** отвечают за слова, содержащие по две различные буквы. Наконец, единственное конечное состояние **S7** соответствует словам, содержащим все буквы алфавита.

Задача 7. (3 балла) Постройте регулярное выражение, распознающее те и только те слова в алфавите **{a, b, c}**, которые содержат все три буквы алфавита.

Решение:
Один из возможных ответов — ***aa*(b(a+b)*c+c(a+c)*b)(a+b+c)*+bb*(a(a+b)*c+c(b+c)*a)(a+b+c)*+cc*(a(a+c)*b+b(b+c)*a)(a+b+c)****.

Первое слагаемое решения, **aa*(b(a+b)*c+c(a+c)*b)(a+b+c)***, описывает слова, начинающиеся с буквы **a**. После некоторого количества букв **a** нам должна встретиться буква **b** или **c**; затем, после, возможно, ещё какого-то количества повторяющихся букв, должна встретиться и третья буква тоже. Заканчивается слово произвольной, возможно пустой, последовательностью букв. Второе и третье слагаемое построены по тому же принципу, но отвечают за слова, начинающиеся на **b** и **c** соответственно.

Задача 8. (3 балла) Постройте машину Тьюринга, распознающую все слова в алфавите **{a, b, c}**, содержащие все три буквы алфавита.
Распознавание происходит следующим образом: если слово распознано, к нему справа дописывается буква **a**, в противном случае — буква **b**.

Решение:

Решение этой задачи полностью аналогично решению задачи номер 6.

s[a]>q1[a]R	q2[c]>q6[c]R	q4[c]>q7[c]R	q6[*]> f[b]N
s[b]>q2[b]R	q2[b]>q2[b]R	q4[b]>q4[b]R	q6[a]>q7[a]R
s[c]>q3[c]R	q2[a]>q4[a]R	q4[a]>q4[a]R	q6[b]>q6[b]R
s[*]> f[b]N	q2[*]> f[b]N	q4[*]> f[b]N	q6[c]>q6[c]R
q1[*]> f[b]N	q3[*]> f[b]N	q5[*]> f[b]N	q7[c]>q7[c]R
q1[a]>q1[a]R	q3[a]>q5[a]R	q5[a]>q5[a]R	q7[b]>q7[b]R
q1[b]>q4[b]R	q3[b]>q6[b]R	q5[c]>q5[c]R	q7[a]>q7[a]R
q1[c]>q5[c]R	q3[c]>q3[c]R	q5[b]>q7[b]R	q7[*]> f[a]N