1 (10 баллов). Напишите генератор формулы colorA(n) в формате DIMACS для задачи о том, что существует 2-раскраска положительных чисел от 1 до n такая, что для любого целочисленного решения $a + b = c$ т.ч. $1 \leqslant a < b < c \leqslant n$ выполняется, что a, b и c не имеют одинаковый цвет.
Подсказка: для каждого целого числа используйте булеву переменную x_i. x_i равно $true$ означает, что i раскрашено в первый цвет, и равно $false$, если i раскрашен во второй цвет.

Решение 1:

Для каждого числа i ($1 \leqslant i \leqslant n$) введем переменную x_i следующим образом:

$$x_i = \begin{cases} true, & \text{если число } i \text{ раскрашено в первый цвет} \\ false, & \text{если число } i \text{ раскрашено во второй цвет} \end{cases}$$

Тогда, будем перебирать числа a, b и c ($1 \leqslant a < b < c \leqslant n$), и в случае, если выполняется, что $a + b = c$ требовать, чтобы выполнялось условие:

$$\overline{(x_a \wedge x_b \wedge x_c) \vee (\overline{x_a} \wedge \overline{x_b} \wedge \overline{x_c})}$$

Поскольку:
$x_a \wedge x_b \wedge x_c$ равно $true \Leftrightarrow$ если a, b и c все имеют первый цвет;
$\overline{x_a} \wedge \overline{x_b} \wedge \overline{x_c}$ равно $true \Leftrightarrow$ если a, b и c все имеют второй цвет;
$(x_a \wedge x_b \wedge x_c) \vee (\overline{x_a} \wedge \overline{x_b} \wedge \overline{x_c})$ равно $true \Leftrightarrow$ если a, b и c имеют одинаковый цвет (первый или второй);
$\overline{(x_a \wedge x_b \wedge x_c) \vee (\overline{x_a} \wedge \overline{x_b} \wedge \overline{x_c})}$ равно $true \Leftrightarrow$ если a, b и c не имеют одинаковый цвет;

Теперь следует преобразовать это к КНФ:

$$\overline{(x_a \wedge x_b \wedge x_c) \vee (\overline{x_a} \wedge \overline{x_b} \wedge \overline{x_c})} = (\overline{x_a} \vee \overline{x_b} \vee \overline{x_c}) \wedge (x_a \vee x_b \vee x_c)$$

Осталось вспомнить правила записи КНФ в DIMACS формате:

- *Комментарий*: **c <comment>**
- *Заголовок*: **p cnf <n> <m>**, где n - количество переменных, m - количество дизъюнктов.
- *Описание дизъюнкта*: номера переменных через пробел, со знаком $-$, если переменная с отрицанием. В конце дизъюнкта всегда 0.
- *Пример КНФ*: $(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3) \wedge (x_2 \vee x_3 \vee \overline{x_4})$.
- *DIMACS формат для примера:*
  c example
  p cnf 4 3
  1 2 -3 0
  -1 -2 3 0
  2 3 -4 0

Генератор формулы colorA(n) находится в программе 01_colorA.py.

2 (10 баллов). Напишите генератор формулы colorB(n) в DIMACS формат для задачи о том, что существует 2-раскраска положительных чисел от 1 до n такая, что для любого целочисленного решения $a^2 + b^2 = c^2$ ($1 \leqslant a < b < c \leqslant n$) выполняется, что a, b и c не имеют одинаковый цвет.

Решение 2:

Данная задача имеет ту же идею решения, что и задача №1. Отличие заключается в том, как будет организован перебор троек чисел a, b и c. Будем перебирать числа a, b и c ($1 \leqslant a < b < c \leqslant n$), и в случае, если выполняется, что $a^2 + b^2 = c^2$ требовать, чтобы выполнялось условие:

$$(\overline{x_a} \vee \overline{x_b} \vee \overline{x_c}) \wedge (x_a \vee x_b \vee x_c)$$

Генератор формулы colorB(n) находится в программе 02_colorB.py.

3 (10 баллов). Используйте указанный выше генератор, чтобы построить формулу colorA(7). Рассмотрим начальные шаги CDCL, в которых $x_1 = 1, x_3 = 1$. Распространение единиц приводит к конфликту. Нарисуйте граф импликации и вычислите первую уникальную точку импликации этого графа.

Решение 3:

Результат генератора из задачи №1 для colorA(7):

```
c colorA(7)
p cnf 7 18
-1 -2 -3 0
1 2 3 0
-1 -3 -4 0
1 3 4 0
-1 -4 -5 0
1 4 5 0
-1 -5 -6 0
1 5 6 0
-1 -6 -7 0
1 6 7 0
-2 -3 -5 0
2 3 5 0
-2 -4 -6 0
2 4 6 0
-2 -5 -7 0
2 5 7 0
-3 -4 -7 0
3 4 7 0
```

Таким образом, мы имеем следующую систему дизъюнктов:
$c_1 = (\neg x_1 \vee \neg x_2 \vee \neg x_3)$
$c_2 = (x_1 \vee x_2 \vee x_3)$
$c_3 = (\neg x_1 \vee \neg x_3 \vee \neg x_4)$

$c_4 = (x_1 \vee x_3 \vee x_4)$
$c_5 = (\neg x_1 \vee \neg x_4 \vee \neg x_5)$
$c_6 = (x_1 \vee x_4 \vee x_5)$
$c_7 = (\neg x_1 \vee \neg x_5 \vee \neg x_6)$
$c_8 = (x_1 \vee x_5 \vee x_6)$
$c_9 = (\neg x_1 \vee \neg x_6 \vee \neg x_7)$
$c_{10} = (x_1 \vee x_6 \vee x_7)$
$c_{11} = (\neg x_2 \vee \neg x_3 \vee \neg x_5)$
$c_{12} = (x_2 \vee x_3 \vee x_5)$
$c_{13} = (\neg x_2 \vee \neg x_4 \vee \neg x_6)$
$c_{14} = (x_2 \vee x_4 \vee x_6)$
$c_{15} = (\neg x_2 \vee \neg x_5 \vee \neg x_7)$
$c_{16} = (x_2 \vee x_5 \vee x_7)$
$c_{17} = (\neg x_3 \vee \neg x_4 \vee \neg x_7)$
$c_{18} = (x_3 \vee x_4 \vee x_7)$

Conflict-driven clause learning (CDCL):

- Изначально, пока не сделали ни одно предположение находимся на нулевом уровне.
- Каждое предположение о значении переменной порождает новый уровень.
- Будем писать $x_i@dl$, если на уровне принятия решений dl мы присвоили переменной x_i значение истина и $\neg x_i@dl$ - если присвоили ложь.
- Вершины графа - переменные, определенные частичной оценкой.
- Из вершины v_i идет ребро в вершину v_j , если переменная v_j оценена в результате $BCP()$ и v_i входит в дизъюнкт-предпосылку c. Это ребро помечается меткой c.
- Если есть конфликт, то ему соответствует вершина **K**. Пусть c - конфликтный дизъюнкт. Тогда к вершине **K** идут ребра от переменных, входящих в c и они помечаются меткой c.
- При возникновении конфликта откатываемся к предпоследнему уровню, эксперименты показывают, что так быстрее.
- *Уникальная точка импликации* – любая вершина графа импликации, которая не является вершиной «конфликт» и которая находится на каждом пути, ведущему от конкретной корневой вершины к конкретной вершине «конфликт».
- *Первая УТИ* – ближайшая к конфликтной вершине уникальная точка импликации.

На рисунке 1 представлен граф импликации.

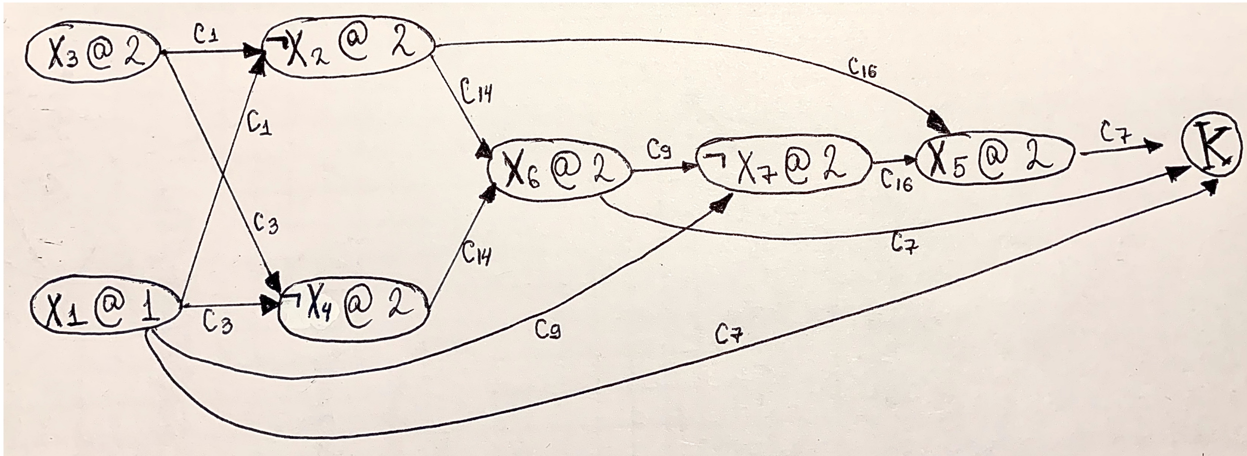

Рис. 1: Граф импликации для задачи 3.

Видим, что:

- Первая УТИ для корневой вершины x_1@1 – вершина x_1@1.
- Первая УТИ для корневой вершины x_3@2 – вершина x_3@2.

4 (10 баллов). Напишите общий генератор DIMACS для головоломок n-судоку, т.е. для пустого поля.
Пояснение: n-судоку – это значит, что дано поле $n^2 \times n^2$, которое разбито на квадраты размера $n \times n$. В каждой строчке, столбце, квадрате может быть не более одного числа от 1 до n^2. Поля 2-судоку и 3-судоку представлены на первом и втором рисунках ниже:

2			
		3	
			1
		2	

2-судоку.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

3-судоку.

Решение 4:

Будем кодировать n-судоку следующим образом $(1 \leqslant i, j, k \leqslant n^2)$:

$$x_{ij}^k = \begin{cases} true, & \text{если в } n\text{-судоку в клетке } i,\, j \text{ стоит число } k \\ false, & \text{если в } n\text{-судоку в клетке } i,\, j \text{ стоит число не } k \end{cases}$$

Таким образом для каждой клетки n-судоку будет n^2 переменных, а всего будет $n^2 \cdot n^2 \cdot n^2 = n^6$ переменных.

Для начала зафиксируем правило расстановки чисел в таблице, не учитывая правил судоку:

1. В каждой клетке n-судоку **должно быть хотя бы одно** значение от 1 до n^2:

$$\phi_1 = \bigwedge_{\substack{1\leqslant i\leqslant n^2\\ 1\leqslant j\leqslant n^2}} \Big(\bigvee_{1\leqslant k\leqslant n^2} x_{ij}^k \Big)$$

2. В каждой клетке n-судоку **должно быть не более одного** значения от 1 до n^2 (***AtMostOne***):

$$\phi_2 = \bigwedge_{\substack{1\leqslant i\leqslant n^2\\ 1\leqslant j\leqslant n^2\\ 1\leqslant k<l\leqslant n^2}} \left(\overline{x_{ij}^k} \vee \overline{x_{ij}^l}\right)$$

Теперь необходимо выполнение условия, что в каждой строчке, столбце и квадрате может быть не более одного числа от 1 до n^2. Отметим, что так как в каждой строке, столбце и квадрате ровно n^2 чисел то достаточно проверить, что них представлены все числа от 1 до n^2 – по принципу Дирихле будет верно, что в них все числа различны.

1. В каждой строчке есть все числа от 1 до n^2:

$$\phi_3 = \bigwedge_{\substack{1\leqslant i\leqslant n^2\\ 1\leqslant k\leqslant n^2}} \Big(\bigvee_{1\leqslant j\leqslant n^2} x_{ij}^k \Big)$$

2. В каждом столбце есть все числа от 1 до n^2:

$$\phi_4 = \bigwedge_{\substack{1\leqslant j\leqslant n^2\\ 1\leqslant k\leqslant n^2}} \Big(\bigvee_{1\leqslant i\leqslant n^2} x_{ij}^k \Big)$$

3. В каждом квадрате есть все числа от 1 до n^2:

$$\phi_5 = \bigwedge_{\substack{0\leqslant i'<n\\ 0\leqslant j'<n\\ 1\leqslant k\leqslant n^2}} \Big(\bigvee_{\substack{1\leqslant i\leqslant n\\ 1\leqslant j\leqslant n}} x_{i'\cdot n+i,j'\cdot n+j}^k \Big)$$

Итоговая формула для генерации n-судоку будет:

$$\phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4 \wedge \phi_5$$

Генератор DIMACS для головоломок n-судоку находится в программе 04_sudoku_gen.py.

5 (5 баллов). Рассмотрим три приведенных поля судоку на рисунке 2. Какое из них можно решить с помощью распространения единицы? Выберите поле, которое можно решить с помощью распространения единицы (unit propagation).

2			
		3	
			1
		2	

2			
		3	
			1
	1		

2			
		3	
			1
	3		

Рис. 2: Рисунок для задачи 5 и 6.

Решение 5:

С помощью распространения единицы можно решить судоку №2. На рисунке 3 можно видеть, что действительно без каких-либо дополнительных предположений с порождением новых слоев в графе CDCL мы смогли получить решение судоку. Для первого и третьего рисунка так не получится, в них придется вносить дополнительные предположения. (Красным цветом обозначены числа, которые нельзя написать из-за ограничей.)

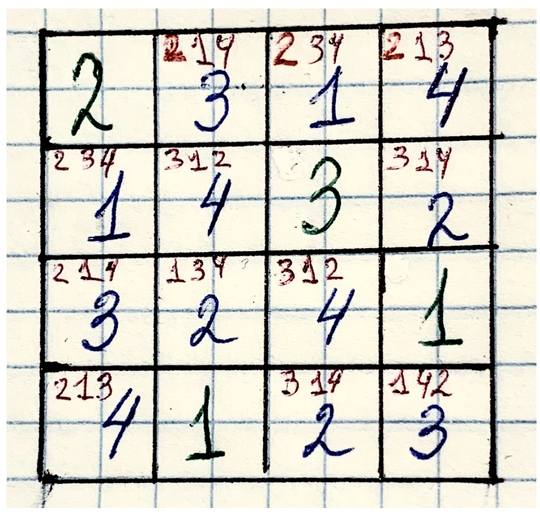

Рис. 3: Решение второго судоку с помощью распространения единицы.

6 (5 баллов). Рассмотрим три приведенных поля судоку на рисунке 2. Какие из них выполнимы? Перечислите все удовлетворяющие оценки полей в виде списка истинных литералов.

Решение 6:

Заметим, что можно модифицировать формулу из задачи №4, добавив условие на уже присутствующие числа в судоку. Полученную формулу в DIMACS формате уже можно подать на вход солверу.

```
-----------------------------------------
Судоку №1:
SAT ✅
2314
4132
3241
1423

SAT ✅
2314
1432
3241
4123

-----------------------------------------
Судоку №2:
SAT ✅
2314
1432
3241
4123

-----------------------------------------
Судоку №3:
UNSAT ❌
```

Рис. 4: Вывод программы **sudoku_solver.py** в задаче №6.

Генератор DIMACS для головоломок n-судоку находится в программе 06_sudoku_solver.py.

На рисунке 4 представлен вывод программы для различных судоку с рисунка 2.

Список истинных литералов очевидно следует из рисунка 4. Например, число 1, находящееся в 3-ей строке и в 4-ом столбце первого судоку соответствует истинному литералу $x^1_{3,4}$.