Программное решение 2ых задач

Курс подготовки к ЕГЭ по информатике 2026

Вступление

Идея решения в переборе всех возможных значений на месте пропусков в таблице, и переборе всех перестановок значений переменных в первой строке таблицы. Перед тем как приступить к задаче подразумевается, что вы, само собой, знаете основы программирования, но помимо основ также рекомендую ознакомиться с:

— Библиотека itertools;

— Словарь;

— all/any.

ФИПИ 885F43

Миша заполнял таблицу истинности функции

$$(x \wedge y) \vee (y \equiv z) \vee w,$$

но успел заполнить лишь фрагмент из трёх различных её строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных w, x, y, z.

				$(x \wedge y) \vee (y \equiv z) \vee w$
1		0	0	0
	1		0	0
1	0	1		0

Определите, какому столбцу таблицы соответствует каждая из переменных w, x, y, z.

В ответе напишите буквы w, x, y, z в том порядке, в котором идут соответствующие им столбцы (сначала буква, соответствующая первому столбцу; затем буква, соответствующая второму столбцу, и т.д.). Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

Решение

Программное решение данной задачи является переборным.

1. Перебираем что может оказаться на месте каждого из пропущенных значений в таблице.
2. Перебираем перестановки из переменных, чтобы определить значения в первой строке.

Для перебора значений пропущенных в таблице используем product. В данной задаче в таблице пропущено 4 значения, поэтому repeat=4.

```
from itertools import *

for v in product("01",repeat=4):
    print(v)
```

Перепишим нашу таблицу заполнив пропуски значениями из v.

```
from itertools import *

for v in product("01",repeat=4):
    a,b,c,d=map(int,v)
    table=[[1,a,0,0,0],
           [b,1,c,0,0],
           [1,0,1,d,0]]
```

Затем проверяем, что все строки уникальны. Для этого перебираем уникальные пары строк, и если хотя бы две совпадут, то такая комбинация из нулей и единиц нам не подходит и мы берем следующую.

```
from itertools import *

for v in product("01",repeat=4):
    a,b,c,d=map(int,v)
    table=[[1,a,0,0,0],
           [b,1,c,0,0],
           [1,0,1,d,0]]
    haveRepeats=False
    for i in range(len(table)):
        for j in range(i+1,len(table)):
            if table[i]==table[j]:
                haveRepeats=True
    if haveRepeats: continue
```

Теперь переберем перестановки из xyzw и сразу составим словарь, где ключи - x, y, z и w, а значения - их позиции.

```
from itertools import *

for v in product("01",repeat=4):
    a,b,c,d=map(int,v)
    table=[[1,a,0,0,0],
           [b,1,c,0,0],
           [1,0,1,d,0]]
    haveRepeats=False
    for i in range(len(table)):
        for j in range(i+1,len(table)):
            if table[i]==table[j]:
                haveRepeats=True
    if haveRepeats: continue
    for n in permutations("xyzw"):
        dic={}
        for i in range(len(n)):
            dic[n[i]]=i
```

Далее проверим все ли строки "хороши". Т.е. все ли строки в результате подстановки дают верное логическое тождество. Если хотя бы в одной из строк это не так, то такая комбинация из a, b, c, d и x, y, z, w нам не подходит. И теперь в каждой строке, для того чтобы выражение выглядело как в условии, перепишим эти значения в соответствующие переменные, а последнее значение строки запишем в переменную f и будем проверять есть ли хотя бы одна строка в которой тождество не сработало. Если такой строки нет, то мы нашли ответ.

```
from itertools import *

for v in product("01",repeat=4):
    a,b,c,d=map(int,v)
    table=[[1,a,0,0,0],
           [b,1,c,0,0],
           [1,0,1,d,0]]
    haveRepeats=False
    for i in range(len(table)):
        for j in range(i+1,len(table)):
            if table[i]==table[j]:
                haveRepeats=True
    if haveRepeats: continue
    for n in permutations("xyzw"):
        dic={}
        for i in range(len(n)):
            dic[n[i]]=i
        good = True
        for line in table:
            x = line[dic["x"]]
            y = line[dic["y"]]
            z = line[dic["z"]]
            w = line[dic["w"]]
            f = line[-1]
            if ((x and y) or (y==z) or w)!=f:
                good = False
                break
        if good:
            print(n)
```

Ну и конечный штрих, чтобы программа выводила ответ в том виде, в котором мы должны вбить его в проверяющую систему, добавим join.

```
from itertools import *

for v in product("01",repeat=4):
    a,b,c,d=map(int,v)
    table=[[1,a,0,0,0],
           [b,1,c,0,0],
           [1,0,1,d,0]]
    haveRepeats=False
    for i in range(len(table)):
        for j in range(i+1,len(table)):
            if table[i]==table[j]:
                haveRepeats=True
    if haveRepeats: continue
    for n in permutations("xyzw"):
        dic={}
        for i in range(len(n)):
            dic[n[i]]=i
        good = True
        for line in table:
            x = line[dic["x"]]
            y = line[dic["y"]]
            z = line[dic["z"]]
            w = line[dic["w"]]
            f = line[-1]
            if ((x and y) or (y==z) or w)!=f:
                good = False
                break
        if good:
            print(''.join(n))
```

P.S. Следите за уникальностью названий переменных. Если идти по латинскому алфавиту для заполнения пропусков в исходной таблице, то очень быстро мы дойдем до f, которую используем для проверки в конце. Конкретно в этом коде это ни на что не повлияет, но лучше стараться такого избегать.