

C/38-159



Олимпиада школьников «Гранит науки» Шифр работы _____
БЛАНК ОЛИМПИАДНОЙ РАБОТЫ

(не заполнять)

ВАРИАНТ № 6

Оценка выполнения олимпиадной работы (заполняется проверяющим)												
Первая проверка	№ задания	1	2	3	4	5	6	7	8	9	10	Σ
	Полученный балл	—	7	7	—	18	23					55
	Σ баллов прописью	Пятдесят пять (55)						Подпись проверяющего		[Signature]		-38
Вторая проверка	№ задания	1	2	3	4	5	6	7	8	9	10	Σ
	Полученный балл	—	4	7	—	18	23					52
	Σ баллов прописью	Пятьдесят два (52)						Подпись проверяющего		[Signature]		

5. Python: ~~MOD = 10 * x * g + 7~~

~~n, r, d = map(int, input().split())~~

~~green = 0~~

~~red = 1~~

~~while n:~~

~~green = (green + red * r)~~

5. Kqz na Python:

MOD = 10 * x * g + 7

n, r, d = map(int, input().split())

g = n - r

green = 0

red = 1

while d:

green = (green + red * g) % MOD

red = (red * r) % MOD

d -= 1

print(red, green)

См. лист 2

185 mod

185

180

ЛИСТ 1 из 5



Вариант № 6

3. Найдите выражение, которое

3 Перенесите логическую схему в виде выражения
 $(x_1 \oplus x_2 \oplus x_3) \& x_4 \& x_3 = 1 \Leftrightarrow$

$$(x_1 \oplus x_2 \oplus x_3) \& x_1 \& x_3 = 1$$

Из этого следует, что $x_1 = 1, x_3 = 1, a$

$$(1 \oplus x_2 \oplus 1) = 1$$

$$0 \oplus x_2 = 1$$

$x_2 = 1 \Rightarrow$ Только 1 набор: $x_1 = 1, x_2 = 1, x_3 = 1$

Подставим его в выражение $x_1 \rightarrow \bar{x}_2 \& x_3 \oplus x_1$

Определим приоритет операций:

$\bar{x}_2$ - отрицание

$\bar{x}_2 \& x_3$ - конъюнкция

$x_1 \rightarrow \bar{x}_2 \& x_3$ - импликация

$x_1 \rightarrow \bar{x}_2 \& x_3 \oplus x_1$ - исключающее или

x_1	x_2	x_3	$\bar{x}_2$	$\bar{x}_2 \& x_3$	$x_1 \rightarrow \bar{x}_2 \& x_3$	$x_1 \rightarrow \bar{x}_2 \& x_3 \oplus x_1$
1	1	1	0	0	0	1

Ответ: 1

2. Нарисует таблицу, в которой ^{номер} столбцы и строки означает количество каток в кучках. Если поле белое - позиция выигрышная, иначе - поле черное. Поле 0,0 закрасит в черный (мой не может сделать ход) см. лист 3

Вариант № 6

Далее будет заполнять таблицу по такому правилу:

Идет по возрастанию номеров строки, в каждой строке по возрастанию номеров столбцов. Так мы не будем проверять квантифицированные клетки.

Посмотрим, если может, клетки сверху, справа, и сверху-справа (ходы $-1,0$; $0,-1$; $-1,-1$ соответственно). Если хотя бы одна из клеток черная, рассматриваемая клетка будет белой, иначе — черной.

20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																				0
																				1
																				2
																				3
																				4
																				5
																				6
																				7
																				8
																				9
																				10
																				11
																				12
																				13
																				14
																				15

$$4 + 3 = 70$$

70 баллов

мемалотт

70

48

-38

Нетрудно заметить, что в черный окрасились те клетки, в которых оба тела были четными. 13 и 9 — оба числа нечетные, значит, это венгерская позиция и победит гном, который ходит первым.

Ответ: Первый гном

Б. Проведем код решения на Python

Будет бинарным поиском перебирать размер робота от 0 до $\min(n, m)$. Будет рекурсивно, не посещая уже посещенные ячейки, проверять все варианты из данной, это работает за 1

лист 3 из 5

Вариант № 6

```

from sys import setrecursionlimit
setrecursionlimit(10000)
nm = map(int, input().split())
a = [None] * n
for i in range(0, n):
    a[i] = input()
def goodgood_pair(i, j):
    global n, m
    return 0 <= i < n and 0 <= j < m
def good(i1, j1, i2, j2):
    global a
    for i in range(i1, i2+1):
        for j in range(j1, j2+1):
            if a[i][j] == '#':
                return False
    return True
def move(i1, j1, i2, j2):
    global m, visited
    visited.append((i1, j1)) visited.add((i1, j1))
    if not good_pair(i1, j1) or not good_pair(i2, j2):
        return False
    if not good(i1, j1, i2, j2):
        return False
    if j2 == 2 * m:
        return True
    res = False
    if (i1+1, j1) not in visited:
        res = res or move(i1+1, j1, i2+1, j2)
    if (i1-1, j1) not in visited:
        res = res or move(i1-1, j1, i2-1, j2)
    if (i1, j1+1) not in visited:
        res = res or move(i1, j1+1, i2, j2+1)
    if (i1, j1-1) not in visited:
        res = res or move(i1, j1-1, i2, j2-1)
    return res

```

Вариант № 6

```

visited = set()
l = 1
r = min(N, M) + 1
while r - l > 1:
    m = (r + l) // 2
    res = False
    for i in range(n):
        res = res or move(i, 0, i + m - 1, j + m - 1)
    if res:
        l = m
    else:
        r = m
    visited = set()
print(l)

```

$$18 + 5 = 238$$

$$188$$

$$238$$