

Задание 19 (Демо 2021) 2 кучи

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежат две кучи камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может добавить в одну из куч (по своему выбору) один камень или увеличить количество камней в куче в два раза. Например, пусть в одной куче 10 камней, а в другой 5 камней; такую позицию в игре будем обозначать (10, 5). Тогда за один ход можно получить любую из четырёх позиций: (11, 5), (20, 5), (10, 6), (10, 10). Для того чтобы делать ходы, у каждого игрока есть неограниченное количество камней.

Игра завершается в тот момент, когда суммарное количество камней в кучах становится не менее 77. Победителем считается игрок, сделавший последний ход, т.е. первым получивший такую позицию, при которой в кучах будет 77 или больше камней.

В начальный момент в первой куче было семь камней, во второй куче – S камней; $1 \leq S \leq 69$.

Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника. В описание выигрышной стратегии не следует включать ходы играющего по этой стратегии игрока, не являющиеся для него безусловно выигрышными, т.е. не являющиеся выигрышными независимо от игры противника.

Известно, что Ваня выиграл своим первым ходом после неудачного первого хода Пети. Укажите минимальное значение S , когда такая ситуация возможна.

Решение:

Решим задачу с помощью шаблона на языке программирования **Python**. Если хотите ознакомиться с аналитическим решением задач на теорию игр, можете посмотреть мои статьи по [19 Заданию](#), [20 Заданию](#), [21 Заданию](#). Но с помощью шаблонов на экзамене решать быстрее и легче.

Введём параметр p , который будет олицетворять позицию игры (ход).

	Начальная позиция	Ход Пети	Ход Вани	Ход Пети	Ход Вани	Ход Пети
p	1	2	3	4	5	6

```
def F(x, y, p):  
    if x + y >= 77 and p==3: return True  
    if x + y < 77 and p==3: return False  
  
    return F(x+1, y, p+1) or F(x*2, y, p+1) or F(x, y+1, p+1) or F(x, y*2, p+1)
```

```
for s in range(1, 70):
    if F(s, 7, 1):
        print(s)
```

Заводим функцию **F**. Она принимает параметры: **x** - количество камней в одной куче, **y** - в другой, **p**-позиция игры.

Дальше описываем победу. Если $x+y \geq 77$ и позиция равна 3 (1 Ход Вани), то возвращаем True, что означает победу.

Если, позиция уже равна 3, но суммарное количество камней меньше, чем должно быть для победы, то возвращаем False (проигрыш).

Если мы не вышли на первых двух условиях, то, значит, продолжаем прокручивать ходы, рекурсивно запускаем функцию **F**.

Т.к. здесь формулировка: "Известно, что Ваня выиграл своим первым ходом после неудачного первого хода Пети.", то между функциями ставим союз ИЛИ (or).

В конце перебираем все возможные значения для **s** через цикл **for**, ищем те значения, которые подходят по условию задачи. Значение **p** всегда увеличиваем на 1.

Ответ: 18

Задание 20 (Демо 2021)

Для игры, описанной в предыдущем задании, найдите два таких значения **S**, при которых у Пети есть выигрышная стратегия, причём одновременно выполняются два условия:

- Петя не может выиграть за один ход;
- Петя может выиграть своим вторым ходом независимо от того, как будет ходить Ваня.

Найденные значения запишите в ответе в порядке возрастания.

Решение:

Легко переделать из прошлой задачи.

```
def F(x, y, p):
    if x + y >= 77 and p==4: return True
```

```

if x + y < 77 and p==4: return False
if x + y >= 77: return False

if p%2==0:
    return F(x+1, y, p+1) and F(x*2, y, p+1) and F(x, y+1, p+1) and F(x, y*2, p+1)
else:
    return F(x+1, y, p+1) or F(x*2, y, p+1) or F(x, y+1, p+1) or F(x, y*2, p+1)

for s in range(1, 70):
    if F(s, 7, 1):
        print(s)

```

Теперь должен выигрывать Петя на своём втором ходе. Поэтому в условиях ставим позицию $p=4$.

Добавляется третье условие. Если кто-то выиграл, но на первых двух условиях мы не вышли из функции, то, значит, выиграл не тот, кто нам нужен, следовательно, возвращаем False.

Здесь вопрос отличается от 19 задания. Здесь Петя должен побеждать при любом ходе соперника, а не при одном неудачном ходе Вани, поэтому добавляется ещё условие.

Для чётных p (это ходы Пети), возвращаем разные ходы через **and**, т.к. он должен побеждать в любом случае.

Для нечётных p (это ходы Вани), возвращаем ходы через **or**.

Ответ:

31	34
----	----

Задание 21 (Демо 2021)

Для игры, описанной в задании 19, найдите минимальное значение S , при котором одновременно выполняются два условия:

– у Вани есть выигрышная стратегия, позволяющая ему выиграть первым или вторым ходом при любой игре Пети;

– у Вани нет стратегии, которая позволит ему гарантированно выиграть первым ходом.

Решение:

Опять используем прошлый шаблон, но немного модернизируем.

```
def F(x, y, p):
    if x + y >= 77 and (p==3 or p==5): return True
    if x + y < 77 and p==5: return False
    if x + y >= 77: return False

    if p%2==1:
        return F(x+1, y, p+1) and F(x*2, y, p+1) and F(x, y+1, p+1) and F(x, y*2, p+1)
    else:
        return F(x+1, y, p+1) or F(x*2, y, p+1) or F(x, y+1, p+1) or F(x, y*2, p+1)

def F1(x, y, p):
    if x>=77 and p==3: return True
    if x<77 and p==3: return False
    if x>=77: return False

    if p%2==1:
        return F1(x+1, y, p+1) and F1(x*2, y, p+1) and F1(x, y+1, p+1) and F1(x, y*2, p+1)
    else:
        return F1(x+1, y, p+1) or F1(x*2, y, p+1) or F1(x, y+1, p+1) or F1(x, y*2, p+1)

for s in range(1, 70):
    if F(s, 7, 1):
        print(s)

print()

for s in range(1, 70):
    if F1(s, 7, 1):
        print(s)
```

Здесь Ваня должен выигрывать либо на первом своём ходе ($p=3$), либо на втором своём ходе ($p=5$).

Т.к. Ваня не должен гарантированно выиграть своим первым ходом, то мы создаём ещё одну функцию **F1**, похожую на основную функцию **F**, которая вычисляет, когда Ваня именно гарантированно выигрывает на своём первом ходе ($p=3$). И, затем, мы из тех чисел, которые получились в первой функции **F**, исключаем числа, которые получились во второй функции **F1**.

В первой функции получилось 30,33, а во второй результатов нет. Получается ответ 30.

Ответ: 30